

# Google Asistan vb. İle Sesli Kontrol Edilen Lamba vb. Çıkış Kontrolü

## Giriş:

Projede insanın bilgisayarla etkileşimini konuşarak gerçekleştirmesini sağlayan bir teknoloji olan Speech Recognition ile sesimizi kullanarak giriş ve çıkışları yöneteceğiz. Bizim sesimizle verdiğimiz komutlara göre sinyal gidecek ve tepkiler izlenecektir.

## Kullanılan Donanım Malzemeleri:

1. 1 adet Raspberry Pi
2. 1 adet USB Ses kartı
3. 1 adet Mikrofon
4. 2 adet LED

## Kullanılan Malzemelerin Özellikleri:

Raspberry Pi 3;

- Broadcom BCM2837 64bit ARMv8 dört çekirdek işlemci
- 1.2GHz
- 1GB RAM
- Dahili WiFi – BCM43143
- Bluetooth 4.1 (Bluetooth Low Energy – BLE)
- 40 Adet GPIO
- 4 Adet USB 2
- 4 uçlu Stereo çıkışı ve Composite video çıkışı
- Full HDMI
- Raspberry Pi Kamera bağlanmasını için CSI kamera portu
- Raspberry Pi 7" dokunmatik ekran için DSI ekran portu
- Micro SD soketi mevcut
- Güncellenmiş güç kartı (2,5A'ye kadar destekliyor.)

USB Ses kartı; Temel olarak ele almak gerekirse ses kartı ses kayıt ve teknolojilerinde ses ile ilgili işlevleri yerine getiren dahili veya harici donanımlardır. Kayıtlarınız sırasında sesleri işleyerek dijital bir hale dönüştürüp kayıt programlarınıza aktarmanıza olanak sağlamaktadır. Bu sayede sesi izleyebilir veya dinleyebiliriz gerekli müdahalelerde bulunarak en doğru sonuca ulaşabilirsiniz. Temel olarak günümüzde 3 tip ses kartı bulunmaktadır bunlar sırası ile USB ses kartı, Firewire ses kartı ve Thunderbolt ses kartı olarak ayrılmaktadır. Aralarındaki fark ise isimlerinden anlaşılabilir olduğu üzere farklı bağlantı noktaları sunarak kullanıcının tercihiyle ilgili olarak değişiklik göstermektedir

Mikrofon;Normal kulaklık mikrofonu kullanılabilir. Bizim komutları söyleyebilmemiz için mikrofona ihtiyacımız vardı.

Led; Lambaları temsil etmesi açısından led kullandık.

## YAPIM AŞAMALARI:

Biz her şeyden önce elimizde malzemeler olmaması nedeniyle uzaktan bağlantı sağlayarak projemizi gerçekleştirdik. Uzaktan bağlantıyı sağlamak için ultraVNC kurulumu yaptık. Sonra bağlantı yaptığımız sunucumuzu terminal üzerinden çeşitli komutlar ile yönetebilmek için putty kurulumu yaptık ve dosya erişimi daha kolay ve rahat olduğu içinde winSCP kurulumu yaptık. Bilgisayarlarımıza sanal makine kurup sanal makinenin içerisine de raspbian işletim sistemini kurarak kodlarımızı bu platform üzerinde çalıştırdık. Kodlarımızı çalıştırmadan önce Speech Recognition kurulumunu terminal ekranı üzerinden gerçekleştirdik.

### Speech Recognition Kurulumu

Konuşma Tanıma (Speech Recognition), insanların cihazlarla etkileşim kurma biçimlerinde devrim yaratan bu gelişimin ardındaki itici teknolojidir. Konuşma Tanıma ile makineler ve uygulamalar kullanıcıların konuşarak ilettikleri komutları anlayabilirler. Teknolojinin konuşarak iletilen talepleri dinleme ve anlama becerileri sayesinde kullanıcılar, cihazlarla tuş ve buton yerine konuşarak etkileşim kurabilir.

İlk olarak Raspberry Pi 'deki Raspbian OS'yi güncel hale getireceğiz. Daha sonrasında **Speech Recognition** kitaplığını yükleyip, kurduğumuz kitaplığı test edeceğiz.

1.ADİM: Raspberry Pi 'deki Raspbian OS'yi güncellemek için aşağıdaki kodu yazıyoruz.

- **sudo apt-get update**

2.ADİM: **Speech Recognition** kitaplığını yüklüyoruz.

- **git clone <http://people.csail.mit.edu/hubert/git/pyaudio.git>**  
**cd pyaudio**  
**sudo python setup.py install**  
**sudo apt-get install libportaudio-dev**  
**sudo apt-get install libportaudio0 libportaudio2 libportaudiocpp0 portaudio19-dev**  
**sudo pip install SpeechRecognition**  
**sudo apt-get install espeak**

3.ADİM: Speech Recognition kütüphanesini aşağıdaki kod ile test edelim.

- **python -m speech\_recognition**

```
pi@raspberrypi:~ $ python -m speech_recognition
ALSA lib confmisc.c:767:(parse_card) cannot find card '0'
ALSA lib conf.c:4528:(_snd_config_evaluate) function snd_func_card_driver return
...
A moment of silence, please...
Set minimum energy threshold to 473.906286399
Say something!
```

-Speech Recognition Kütüphanesinin Test Görüntüsü-

Uygulama boyunca yazdığımız kodlara değinecek olursak;

### PYTHON KODU

```
import RPi.GPIO as GPIO
import time
import speech_recognition as sr

# Raspberry Pi kartı pin numaralarını kullanmak için
GPIO.setmode(GPIO.BOARD)
# GPIO çıkış kanalını ayarladık.
GPIO.setup(7, GPIO.OUT)
GPIO.setup(8, GPIO.OUT)

r = sr.Recognizer()
m = sr.Microphone()

def listen():
    try:
        print("Sessizlik, lutfen...")
        with m as source:
            r.adjust_for_ambient_noise(source)
        #ortam gürültü seviyeleri için için 1 saniye dinleyin.

        while True:
            with m as source: #ses kaynağı olarak varsayılan
mikrofonu kullan
                audio = r.listen(source)
                print("Got it! Now to recognize it...")
                try:
                    print("Bir sey soyle!")
                # Google Speech Recognition kullanarak konuşmayı tanıyoruz.
                value = r.recognize_google(audio)
```

```

# unicode karakterleri standart çıktıya doğru şekilde yazdırmak
için burada bazı özel işlemlere ihtiyacımız var.
    if str is bytes: # Python'un bu sürümü, dizeler için
        bayt kullanır (Python 2)
        print(u"You said {}".format(value).encode("utf-8"))
        return value
    else: # Python'un bu sürümü, dizeler için unicode
        kullanır (Python 3+)
        print("You said {}".format(value))
        if value == "can you open bedroom":
            print("Yes, i can ;)")
            GPIO.output(7,True)
        if value == "can you close bedroom":
            print("Yes, i can ;)")
            GPIO.output(7,False)
        if value == "can you open restroom":
            print("Yes, i can ;)")
            GPIO.output(8,True)
        if value == "can you close restroom":
            print("Yes, i can ;)")
            GPIO.output(8,False)
        if value == "can you open everything":
            print("Yes, i can ;)")
            GPIO.output((7,8),True)
        if value == "can you close everything":
            print("Yes, i can ;)")
            GPIO.output((7,8),False)

except sr.UnknownValueError:
    print("Oops! Didn't catch that")
except sr.RequestError as e:
    print("Uh oh! Couldn't request results from Google
Speech Recognition service; {}".format(e))
except KeyboardInterrupt:
    pass

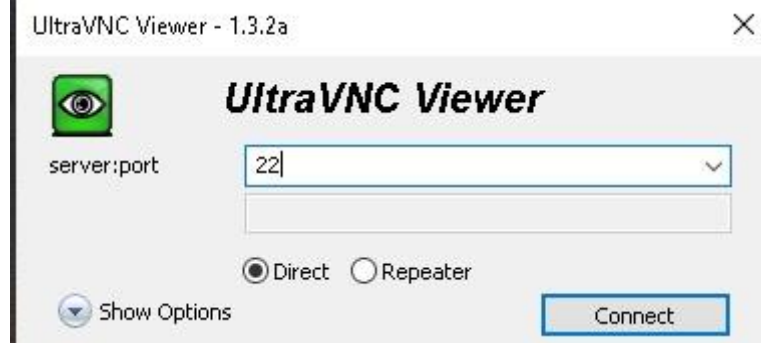
```

## UZAK BAĞLANTI İÇİN YAPTIĞIMIZ ADIMLAR

1.ADIM: Global Protect üzerinden merkezi kimlik bilgilerimiz ile giriş yapıp uygulamayı kurup bağlantısını sağladık.



2.ADIM: **UltraVNC**, bilgisayarınızı uzaktan yönetmenizi sağlayan VNC programlarından biri. **UltraVNC** ile uzakta olan herhangi bir bilgisayarınıza bağlanarak yapmanız gereken tüm işlemleri yapabilirsiniz. Bizde **UltraVNC** uygulamasını kullanarak bağlantıyı gerçekleştirdik. İlk önce port numarasını giriyoruz daha sonrasında bağlantıyı başarılı bir şekilde gerçekleştiriyoruz.

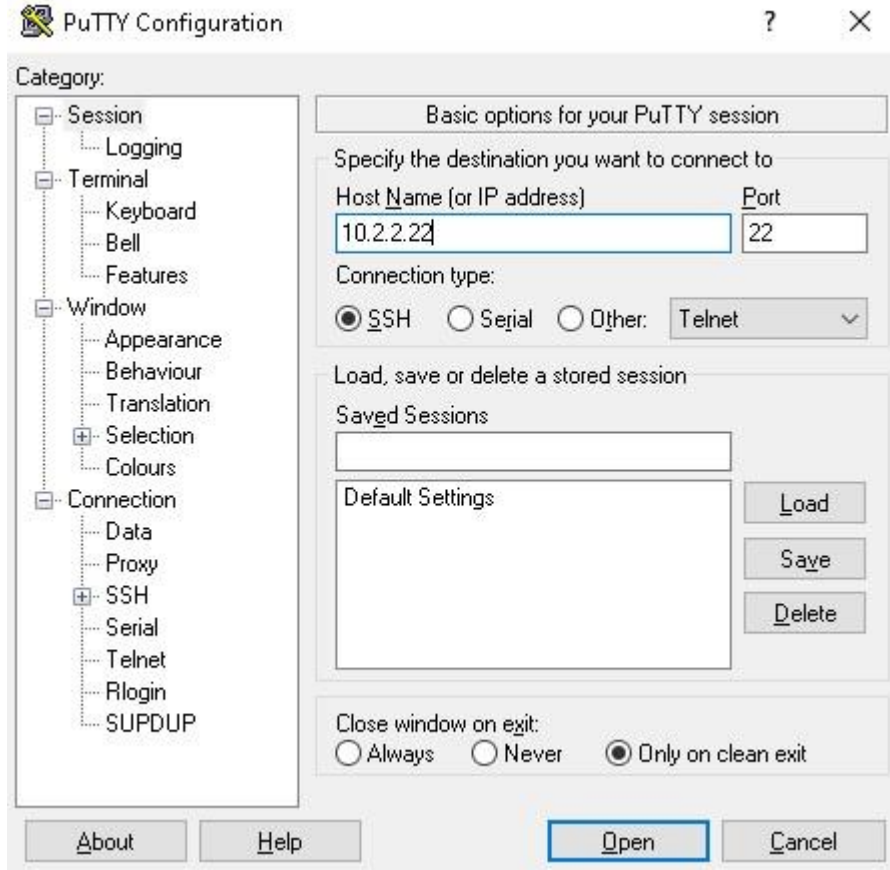


-Resim1-

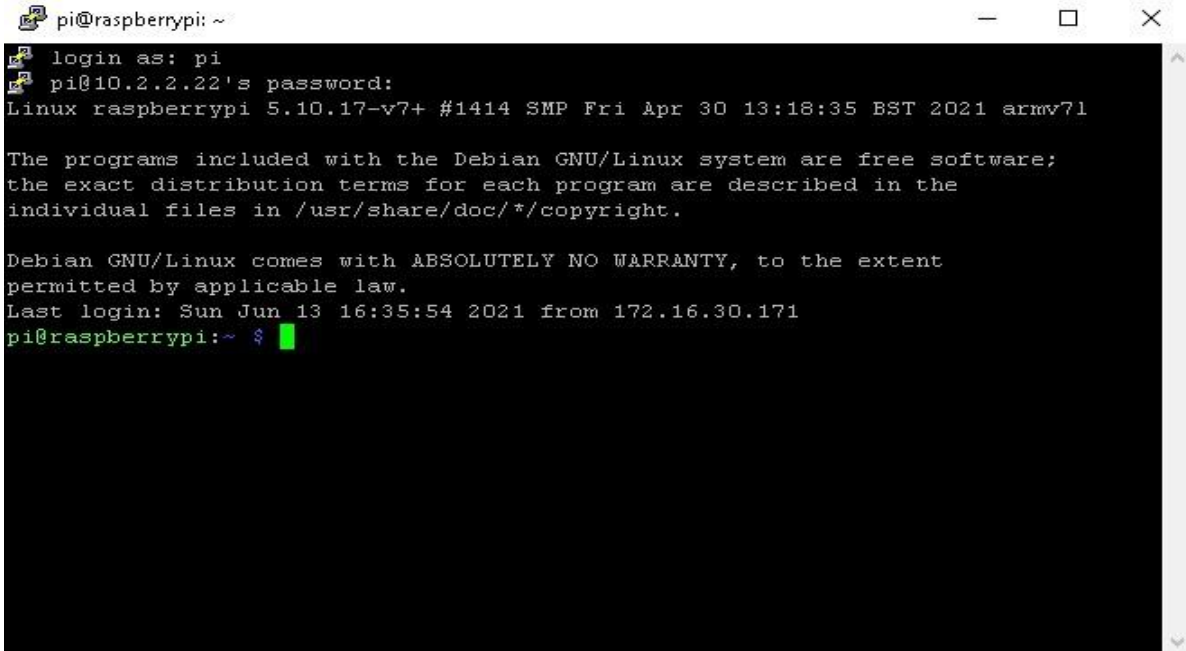


-Resim2-

3.ADIM: **UltraVNC**'den sonra PuTTY programına giriyoruz. İlk önce Host Name yani IP adres kısmını doğru bir şekilde girip OPEN diyoruz. Sonrasında ekrana gelen pi@raspberrypi terminaline Login girişini yapıyoruz. Ve kodu yazar hale gelmiş bulunuyoruz.



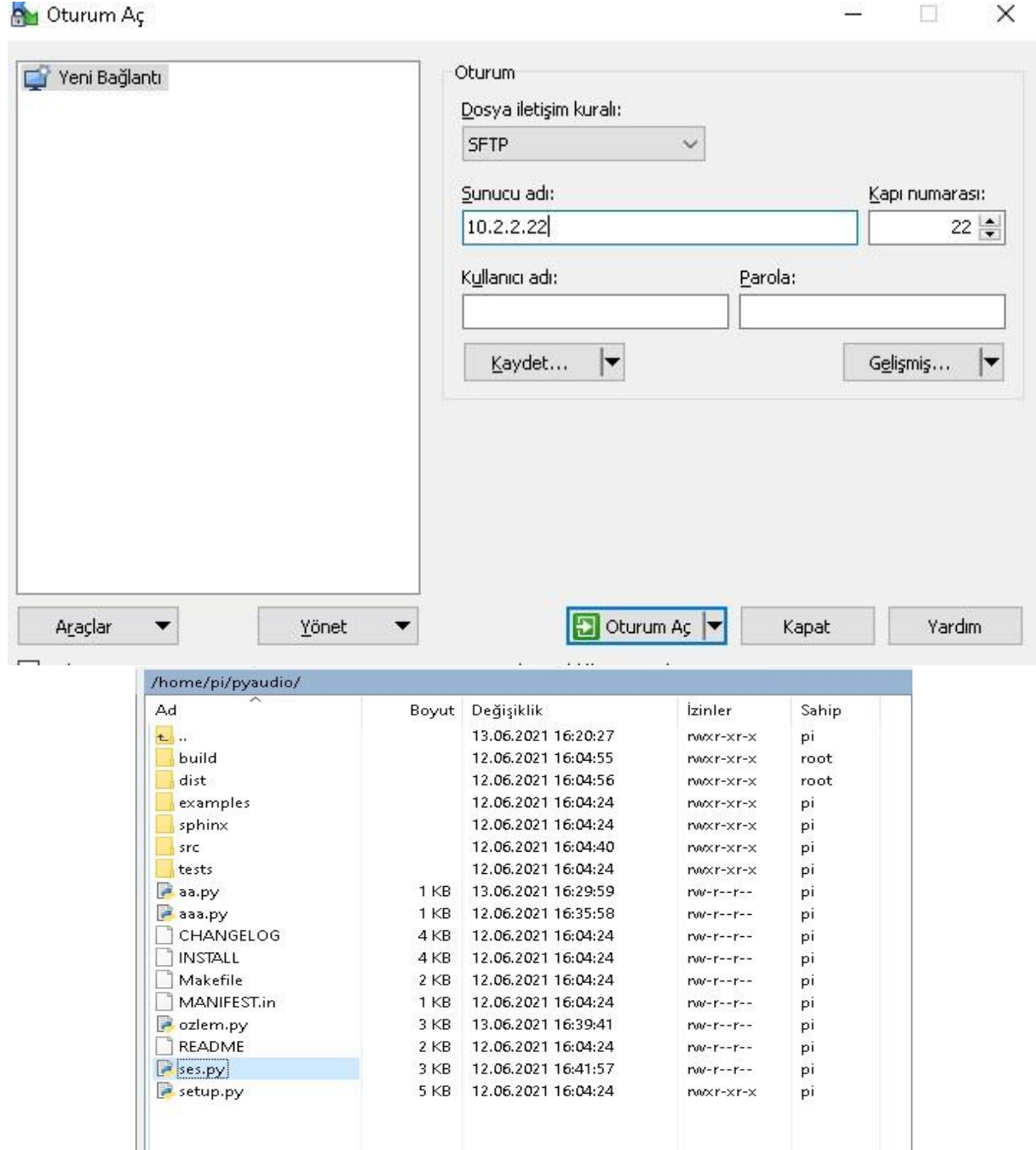
-PuTTY Configuration-



-Raspberrypi Terminal Kısımı-

4.ADIM:Bizim kodumuzun yazılı olduđu Python dosyasını uzak bağlantı sağladığımız bilgisayarda çalıştırmamız için kod dosyasını aktarmamız lazım bunun içinde WinSCP uygulamasını kullanıyoruz. WinSCP, WinSCP , **FTP** dosyalarına erişim sağlamak üzerine inşa edilmiş bir programdır. Asıl amacı güvenli bir şekilde dosyalarımızı sunucuya aktarmaktır.

Öncelikle Sunucu adını, kullanıcı adı ve şifre bilgilerini girdikten sonra Oturum aç diyoruz ve programa giriş yapıp aktarım yapmaya hazır hale getirmiş oluyoruz.



-Dosya aktarımından sonraki kısım-

Aktarımı tamamladıktan sonra Raspberry pi terminalinden kodu çalıştıracak komutu yazdıktan sonra projemiz çalışmaktadır.

Projemimizi çalıştırmak için:

**Cd pyaudio**

**Python ses.py**