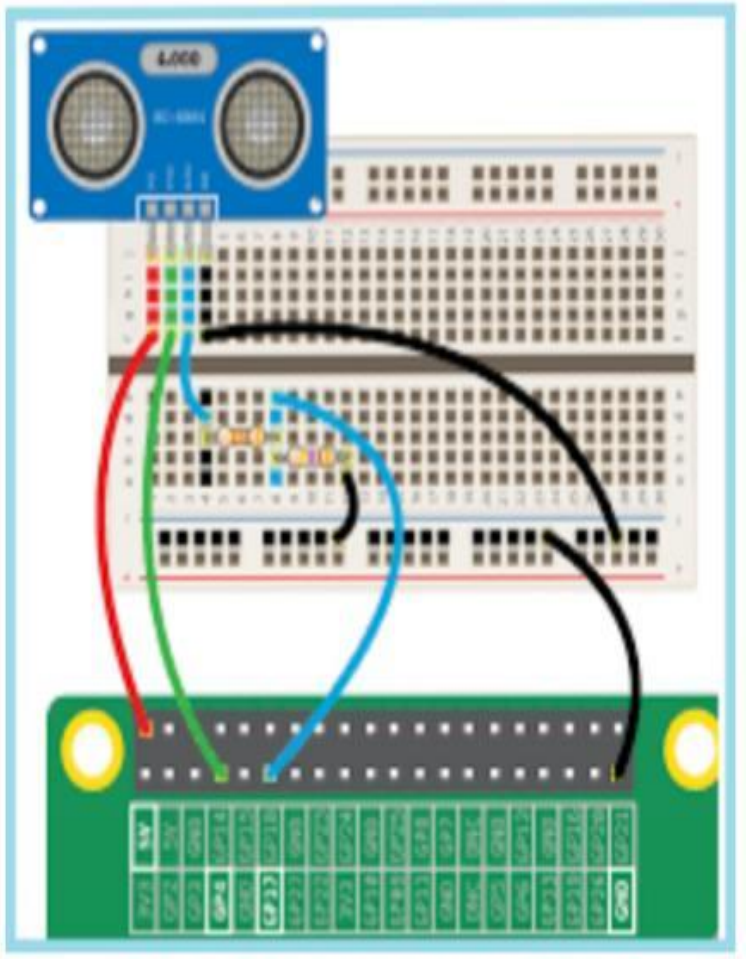
SMART PARKING

TEAM MEMBERS:
S.Roshini vinnarasi-821021104037)
G.Atchaya- 821021104012)
A.Thoumiya- 821021104050)
A.Joy Ramila -821021104025)
D.Aarthi-821021104002)

Raspberry Pi Ultrasonic Sensor Interface Tutorial

In this article, we discuss the interfacing of ultrasonic sensors with Raspberry Pi.



ultrasonic interfacing with Raspberry Pi GPIO. In our earlier article, we talked about the Raspberry Pi GPIO pinouts. If you haven't gone through it, I request you take a look, here is the link. In this article, we will discuss the interfacing of ultrasonic sensors with Raspberry Pi and a little description of the "Python code for ultrasonic sensor".Using

PMUsing Ultrasonic Sensors With Raspberry Pi GPIO

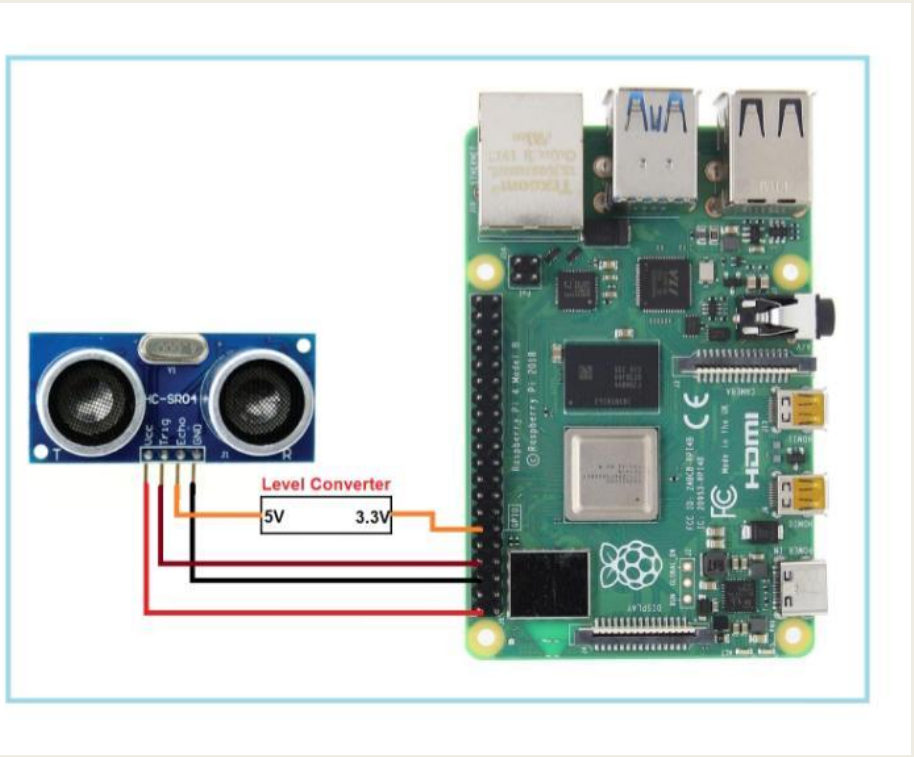
- As you know ultrasonic sensors are used for distance measurement purposes.
- Typically, we use ultrasonic sensors in obstacle prevention robots and DIY radar projects.
- Similar sensors like IR sensor, LIDAR, sonar sensor are also available in the market which can be used for similar purposes but in this article, we are not talking about them.
- I will write about them in my upcoming articles.

How Does the Ultrasonic Sensor Works?

- As you are from the DIY community, I'm pretty sure you know how an ultrasonic sensor works.
- If you know about it then you can skip this part of this blog and go to the coding section. But if you are new to this then you can continue this section.So, ultrasonic sensor work is somewhat similar to radar sensor work.
- In radar, the transmitter generates radio waves (sound waves), an electromagnetic wave that travels in the air and returns when it hits an object in their path. The distance is then calculated using a simple high- school formula which is given below.
- DistanceTime xSpeed

Interfacing Of The Ultrasonic Sensor With The Raspberry PI 4 GPIO

- As discussed earlier, ultrasonic sensors have a transmitter (Trigger) that can be used to transmit infrared sound waves and have a receiver that receives reflected sound waves (ECHO pin).
- And it has power pins which are VCC and GND.
- The following picture shows the pinout of the ultrasonic sensor and its interface with the Raspberry Pi.

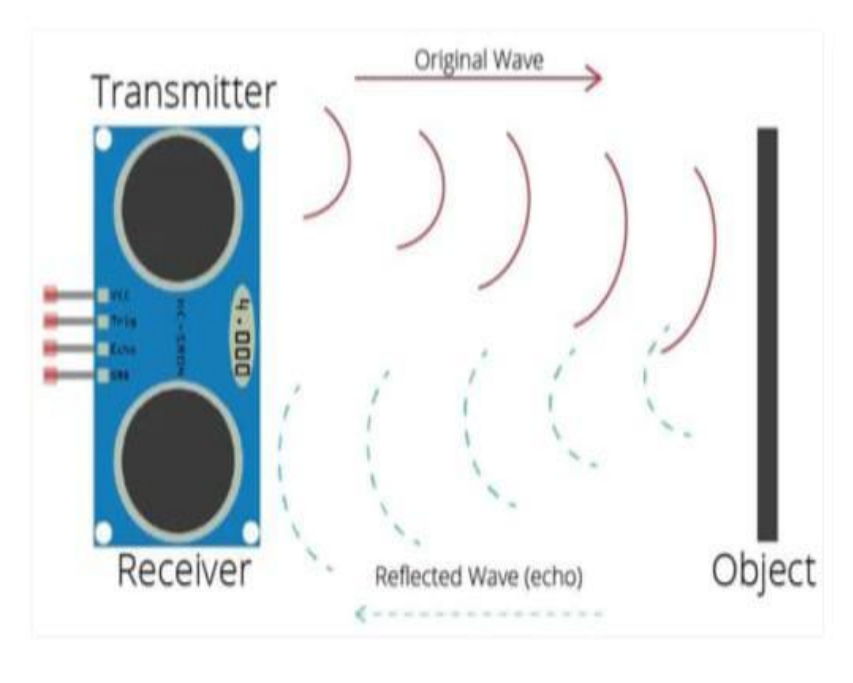


Components that you will require:

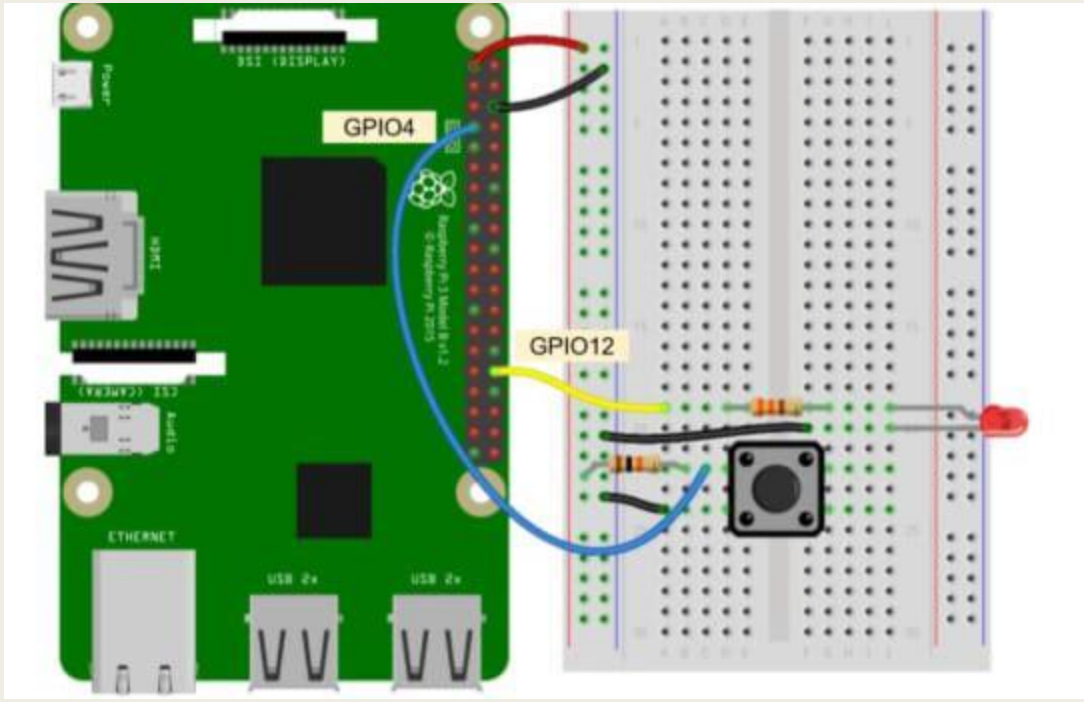
1. *Voltage Level Converter*
2. *Ultrasonic Sensor*
3. *3.5v 3Amp Raspberry Pi Power Adapte*
4. *Raspberry Pi 4*

How Ultrasonic Sensors Work?

- An ultrasonic sensor is a sensor that transmits ultrasonic sound waves and receives them back to calculate the distance from the object.
- It is used in lots of applications like obstacle avoidance robots, measuring liquid in bottles, etc.
- It's similar to SONAR technology used in ships to calculate the de ocean.

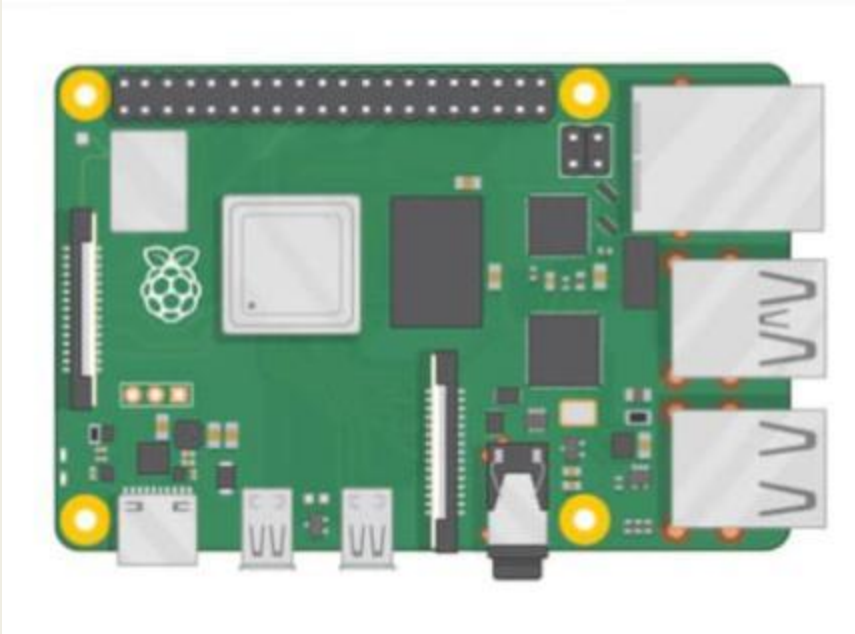


Connecting directly to the Raspbeny Pi

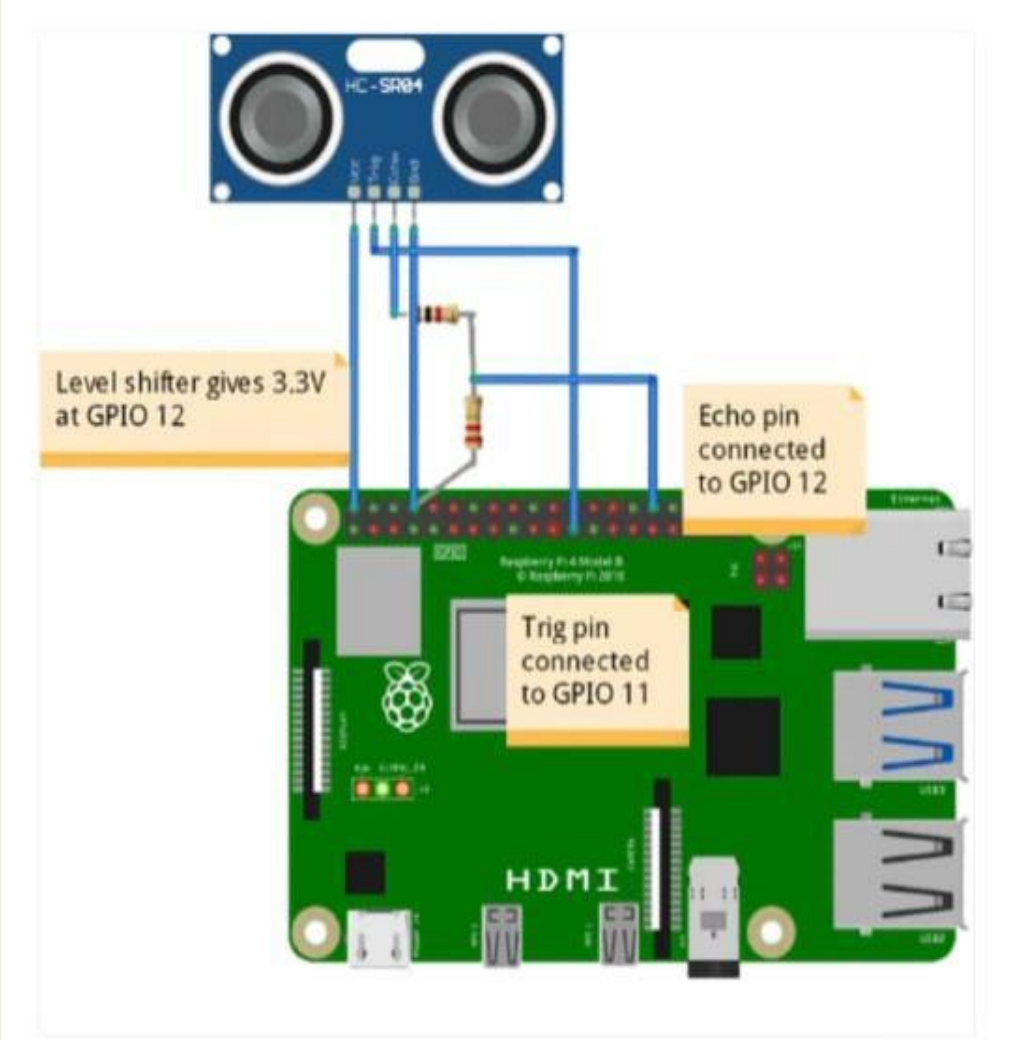


Interfacing HC-SR04 with Raspberry Pi

- In this project we will be using the following
- components:
- 1. Raspberry Pi 4 B
- 2. HC-SR04 ultrasonic sensor module
- 3. 1K,2K Resistors
- 4. Jumper cables
- 5. Breadboard
- 6. Rasberry Pi official power supply adapter



Raspberry Pi respectively



Raspberry pi python script codeing

```
import RPi.GPIO as GPIO
 import time
import requests
# Define GPIO pins for ultrasonic sensors
TRIG = 23
ECHO = 24
# Set GPIO mode and initialize pins
GPIO.setmode(GPIO.BCM)
GPIO.setup(TRIG, GPIO.OUT)
GPIO.setup(ECHO, GPIO.IN)
# Function to measure distance using ultrasonic sensor
def measure_distance():
    GPIO.output(TRIG, True)
```

```
    time.sleep(0.00001)
    GPIO.output(TRIG, False)

    pulse_start = time.time()
    while GPIO.input(ECHO) == 0:
        pulse_start = time.time()

    pulse_end = time.time()
    while GPIO.input(ECHO) == 1:
        pulse_end = time.time()

    pulse_duration = pulse_end - pulse_start
    distance = (pulse_duration * 34300) / 2  # Speed of sound = 343 m/s
    return distance
```

```
# Main loop to collect data and send it to the cloud
Try:
    while True:
        distance = measure_distance()
        if distance < 20:  # Adjust 0 threshold for parking space occupancy
            # Send data to the cloud or mobile app server
            # Replace the following line with your actual data sending code
            response = requests.post('your_server_url', data={'distance': distance})
            print(f"Distance: {distance} cm - Data Sent: {response.status_code}")
            time.sleep(1)  # Adjust the time interval as needed
except KeyboardInterrupt:
    GPIO.cleanup()
```

Conclusion

- In this article, we learned the interfacing of ultrasonic sensors with Raspberry Pi GPIO 4 pins.
- If you liked this article or have any doubt then let me know in the comment section.
- In our next article, we will discuss on the relay module and its interfacing with the raspberry GPIO 4. Stay Tuned!!