

# smart parking

## development part1.

### Hardware Setup:

Connect the ultrasonic sensors to the Raspberry Pi's GPIO pins. Typically, ultrasonic sensors have 4 pins: VCC (power), GND (ground), TRIG (trigger), and ECHO (echo). Connect them accordingly.



### Install Required Libraries:

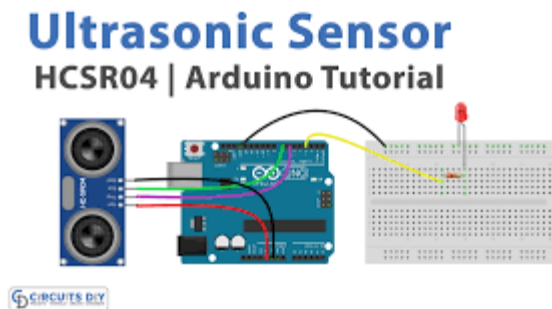
Install necessary python librarise for interacting with the ultrasonic sensors.You can use a library like 'RPi.GPIO' for GPIO control and interfacing with the ultrasonic sensor.



### Ultrasonic Sensor Code:

Write a Python script to read data from the ultrasonic sensor. Initialize the

GPIO pins for trigger and echo. Send a short pulse on the trigger pin to start the ultrasonic ranging module. Measure the duration for the sound wave to bounce back (received on the echo pin) Calculate the distance based on the duration.



Here's a basic example of how the script might look:

```
python
```

```
import RPi.GPIO as GPIO
```

```
import time
```

```
# Set the GPIO pins for trigger and echo
```

```
TRIG_PIN = 23
```

```
ECHO_PIN = 24
```

```
# Setup GPIO
```

```
GPIO.setmode(GPIO.BCM)
```

```
GPIO.setup(TRIG_PIN, GPIO.OUT)
```

```
GPIO.setup(ECHO_PIN, GPIO.IN)
```

```
try:
```

```
while True:
```

```
    # Trigger the ultrasonic sensor
```

```
    GPIO.output(TRIG_PIN, True)
```

```
    time.sleep(0.00001)
```

```
    GPIO.output(TRIG_PIN, False)
```

```
    # Wait for the echo response
```

```
    while GPIO.input(ECHO_PIN) == 0:
```

```
        pulse_start = time.time()
```

```
    while GPIO.input(ECHO_PIN) == 1:
```

```
        pulse_end = time.time()
```

```
    # Calculate distance
```

```
    pulse_duration = pulse_end - pulse_start
```

```
    distance = pulse_duration * 17150
```

```
    distance = round(distance, 2)
```

```
    print(f"Distance: {distance} cm")
```

```
    time.sleep(1)
```

```
except KeyboardInterrupt:
```

GPIO.cleanup()