Ultrasonic Sensor Documentation - Raspberry Pi 5 Integration

El Mehdi Adnani Kadmiri

July 17, 2025

1 Description

The Ultrasonic Sensor (e.g., HC-SR04) uses ultrasonic sound waves to measure the distance between the sensor and an object. It sends out a sound pulse at 40kHz and measures the time taken for the echo to return. This time is then used to calculate distance using the speed of sound.

2 Applications

Ultrasonic sensors, such as the HC-SR04, are widely used in real-world scenarios where accurate, non-contact distance measurement is required. Their ability to detect objects using sound waves makes them ideal for various fields, including:

- **Obstacle Detection:** Used in robotics and automation systems to detect and avoid objects.
- **Parking Assistance:** Commonly integrated into car bumpers to help drivers detect nearby obstacles while parking.
- **Liquid Level Monitoring:** Helps monitor fluid levels in tanks without direct contact, preserving sensor lifespan and hygiene.
- **Smart Waste Management:** Measures fill levels in garbage bins and sends data to optimize collection routes.
- **Industrial Automation:** Detects presence or absence of components on assembly lines to trigger actions.

3 Working Principle

The sensor has two main pins:

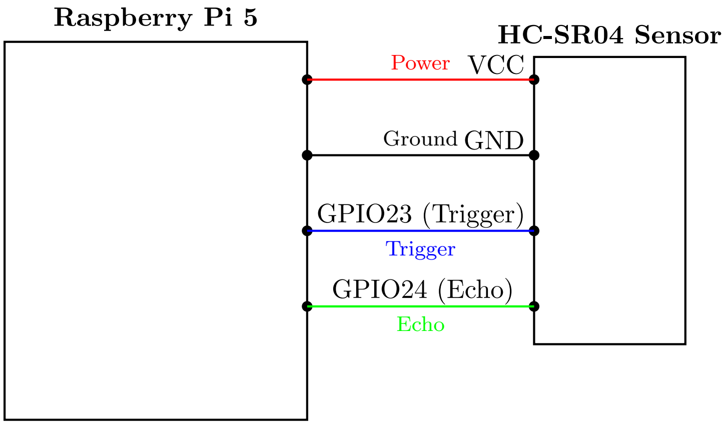
- **Trigger (TRIG)**: Sends an ultrasonic burst (10µs pulse).
- **Echo (ECHO)**: Receives the reflected wave and outputs a pulse proportional to distance.

Using the time between sending and receiving, distance is calculated with:

$$\text{Distance (cm)} = \frac{\text{Time (µs)} \times 0.0343}{2}$$

4 Wiring Diagram

Ultrasonic Pin	Raspberry Pi Pin	Function
VCC	5V	Power
GND	GND	Ground
TRIG	GPIO23	Output Trigger Signal
ECHO	GPIO24	Input Echo Signal



5 Libraries Used

Python: RPi.GPIO

- Setup with: `import RPi.GPIO as GPIO`, `import time`
- Configure mode: `GPIO.setmode(GPIO.BCM)`
- Trigger pulse: `GPIO.output(TRIG, True)`
- Read echo: `GPIO.input(ECHO)`

C: wiringPi

- Initialize GPIO: `wiringPiSetupGpio();`
- Set pins: `pinMode(TRIG, OUTPUT)`, `pinMode(ECHO, INPUT)`
- Control and read: `digitalWrite()`, `digitalRead()`, `micros()`

6 Python Code Example

```python
import RPi.GPIO as GPIO
import time

TRIG = 23
ECHO = 24

GPIO.setmode(GPIO.BCM)
GPIO.setup(TRIG, GPIO.OUT)
GPIO.setup(ECHO, GPIO.IN)

GPIO.output(TRIG, False)
time.sleep(2)

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

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

```
pulse_start = time.time()

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

pulse_duration = pulse_end - pulse_start
distance = (pulse_duration * 34300) / 2

print(”Distance:␣%.2f␣cm” % distance)
GPIO.cleanup()
```

7 C Code Example

```
#include <wiringPi.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>

#define TRIG 23
#define ECHO 24

long getMicroseconds() {
        struct timeval tv;
        gettimeofday(&tv, NULL);
        return tv.tv_sec * 1000000 + tv.tv_usec;
}

int main(void) {
        if (wiringPiSetupGpio() == -1)
        return 1;

        pinMode(TRIG, OUTPUT);
        pinMode(ECHO, INPUT);

        digitalWrite(TRIG, LOW);
        delay(500);

        digitalWrite(TRIG, HIGH);
        delayMicroseconds(10);
        digitalWrite(TRIG, LOW);

        while (digitalRead(ECHO) == LOW);
        long start = getMicroseconds();

        while (digitalRead(ECHO) == HIGH);
        long end = getMicroseconds();

        long duration = end - start;
        float distance = duration * 0.0343 / 2;

        printf(”Distance:␣%.2f␣cm\n”, distance);
        return 0;
}
```

8 Conclusion

The Ultrasonic Sensor provides accurate, non-contact distance measurement and is widely used in robotics. With basic GPIO and timing functions, it integrates well with the Raspberry Pi using both Python and C.