



CSE 461

Lab 2

Name of the Experiment: Measuring distance using an ultrasonic sensor

Name: Sheikh Alima Mahbub

ID: 20101517

Section: 08

Group: 05

Objective: In this lab report, we will be discussing the process of measuring distance using an ultrasonic sensor with the help of a Raspberry Pi. Ultrasonic sensors are a common type of sensor that is used for measuring distances by sending out sound waves and measuring the time it takes for them to bounce back. Raspberry Pi is a small, affordable computer that can be used for various purposes such as programming, robotics, and education. This lab aims to demonstrate how to use an ultrasonic sensor with Raspberry Pi to measure distance accurately.

Components required for the setup:

- Raspberry Pi
- Ultrasonic Sensor (HC-SR04)
- Breadboard
- Jumper Wires
- 1k and 1.5k resistor
- MicroSD Card
- USB Cable
- Monitor, Keyboard, and Mouse

The image shows a Raspberry Pi 4 in a red case, connected to a breadboard circuit. The breadboard contains a 555 timer, resistors, and a motor. Wires connect the Pi's GPIO pins to the breadboard components. The breadboard is populated with a 555 timer, several resistors, and a small motor. The Raspberry Pi is connected to the breadboard via a black USB cable and a red power cable. The breadboard is connected to a power source via a black power cable and a red power cable. The breadboard is connected to a motor via a black power cable and a red power cable. The breadboard is connected to a motor via a black power cable and a red power cable.

Code:

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

TRIG = 21
ECHO = 20

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

def distance():
    GPIO.output(TRIG, False)
    time.sleep(0.5)
    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()
    while GPIO.input(ECHO)==1:
        pulse_end = time.time()
    pulse_duration = pulse_end - pulse_start
    distance = pulse_duration * 17150
    distance = round(distance, 2)

    return distance
While True:

    print(distance())

GPIO.cleanup()
```

Results: The code provided is a Python script that measures distance using an ultrasonic sensor connected to a Raspberry Pi. It calculates the distance based on the time it takes for the ultrasonic waves to bounce back.

When the code is executed, it will measure the distance using the ultrasonic sensor and print the result to the console. The measured distance will depend on the environment and the accuracy of the ultrasonic sensor.

We got the distance around 4cm to 5cm of an object from the ultrasonic sensor.

Discussion: By using a voltage divider circuit, the resistors proportionally divide the voltage from the Echo pin, reducing it to a safe level of 3V approximately that the GPIO pin can handle. The specific resistor values (1k and 1.5k) are chosen to achieve the desired voltage division ratio.

This arrangement ensures that the voltage level received by the Raspberry Pi's GPIO pin from the ultrasonic sensor's Echo pin remains within the safe operating range, protecting the GPIO pin from potential damage.