ID	**NAME**
Syed Ashik Mahamud	20301124
Nisharga Nirjan	20101020
Sabbir Hossain Shaon	20301147
Md. Sajid Ullah Sohan	20301129

Experiment No: 02

Group No: 02

Section: 07

Spring 2023

Name of the Experiment: Measuring distance using ultrasonic sensor.

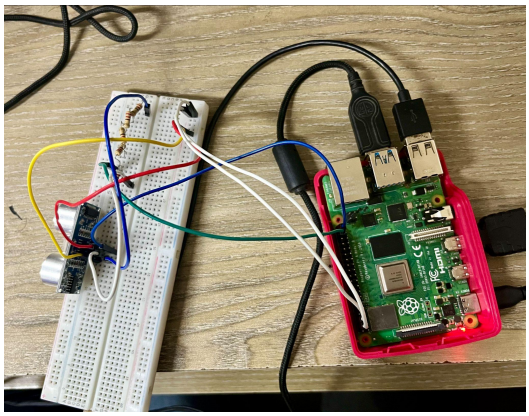
Objective:

This experiment aims to measure distance using ultrasonic sensor by using a switch on the Raspberry Pi 4. The user writes code to detect switch activation, allowing for measure the distance of the object.

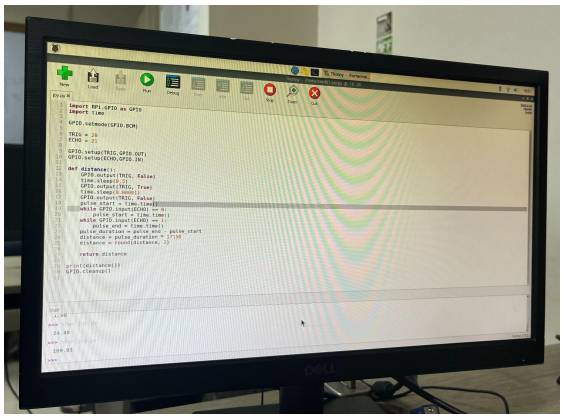
Equipment:

1. Raspberry Pi
2. Ultrasonic Sensor (HC-SR04)
3. Breadboard
4. Jumper Wires
5. 1k and 1.5k resistor
6. MicroSD Card
7. USB Cable
8. Monitor, Keyboard, and Mouse (Optional)

Experimental Setup:



Circuit design



Code and distance result output

Code:

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

TRIG = 20
ECHO = 21

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
print(distance())
GPIO.cleanup()
```

Result:

To conduct the experiment, the ultrasonic sensor is connected to the Raspberry Pi 4 via a breadboard and jumper wires. The 1K and 1.5K resistors may be used as voltage dividers to ensure that the sensor's input voltage does not exceed the Raspberry Pi's maximum input voltage. Though we use five 220 ohm resistor for conducting the experiment. The code used will typically involve triggering the sensor to send out an ultrasonic pulse and measuring the time it takes for the pulse to bounce back to the sensor. This time can then be used to calculate the distance to the object that reflected the pulse. Overall, the experiment is typically successful and can provide accurate distance measurements using the ultrasonic sensor and Raspberry Pi 4 setup.

Conclusion & Discussion:

The experiment demonstrates the accuracy and reliability of using an ultrasonic sensor connected to a Raspberry Pi for measuring distances. Through this experiment, we can conclude that the ultrasonic sensor (HC-SR04) is a reliable and accurate sensor for distance measurements. The Raspberry Pi 4, with its processing power and connectivity, provides a flexible platform for integrating and analyzing sensor data. The breadboard, jumper wires, and resistors are simple yet effective components for building a reliable sensor circuit. Overall, the experiment highlights the usefulness of the ultrasonic sensor and Raspberry Pi 4 combination in distance measurement applications and opens up possibilities for further exploration and innovation in this field.

Explain the following questions:

1) **Why are the resistors used?**

The resistors (we used five 220 ohm resistors) used on the circuit are likely used as voltage dividers to ensure that the input voltage to the ultrasonic sensor (HC-SR04) does not exceed the maximum input voltage of the Raspberry Pi. The HC-SR04 ultrasonic sensor typically operates at 5V, while the Raspberry Pi's GPIO pins operate at 3.3V. If the sensor's input voltage is too high, it can potentially damage the Raspberry Pi's GPIO pins. Therefore, a voltage divider circuit can be used to lower the input voltage to a safe level. A voltage divider circuit consists of two resistors in series. The voltage across each resistor is proportional to its resistance value. By choosing appropriate resistor values, we can distribute the voltage across the circuit in a certain ratio. In this case,the resistors are used in a voltage divider configuration to provide a 3:5 ratio, which reduces the 5V input voltage to a safe level of 3.3V for the Raspberry Pi's GPIO pins. In summary, the resistors are used as voltage dividers to ensure that the input voltage to the ultrasonic sensor is within the safe range for the Raspberry Pi's GPIO pins.