BRAC
UNIVERSITY

Inspiring Excellence

CSE461 Lab Report

Lab Number: 2

Title: Interfacing Sonar Sensor with Raspberry Pi.

Group Number: 4

Section: 4

Group Members:

1. Anika Ahmed, ID: 21101029
2. Nafis Chowdhury, ID: 21101034
3. Moinul Haque, ID: 21101186
4. Yeamin Hossain Shihab, ID: 21101174
5. Mirza Fahad Bin Kamal, ID: 20101399

1. Description:

This lab report's focus is to investigate the process of connecting ultrasonic sensors to a Raspberry Pi in order to comprehend and utilize the sensor's distance measurement capabilities. We initially assembled all of our equipment and noted the connections we needed to make. The ultrasonic sensor has four distinct pins: ground, TRIG, ECHO, and power. We linked the TRIG pin to the GPIO21 pin and the power pin to the Raspberry Pi's 5V voltage source. We linked the GPIO20 pin to the sensor's ECHO pin by using two 220 ohm resistors in series. After that, we connected the sensor's GND and Raspberry Pi's GND by utilizing three resistances in series from the GPIO20's beginning point. In total five 220 ohm resistors were used to create a 2:3 parallel connection.

The TRIG pin, which is the output pin, will produce some pulses. The input pin is the ECHO pin, which will receive the start and end times. We will first have to wait a bit for the sensor to stabilize. Then, the sensor will transmit 8 pulses in 10-microseconds using the TRIG pin. Following that, we'll keep an eye on the ECHO signal. We will begin counting down in a loop when the ECHO pin is low, and we will record the start time when it breaks (i.e., when the ECHO becomes 1). After then, the ECHO will rise, and once more, we'll capture the end time right before it breaks (when the ECHO becomes 0).

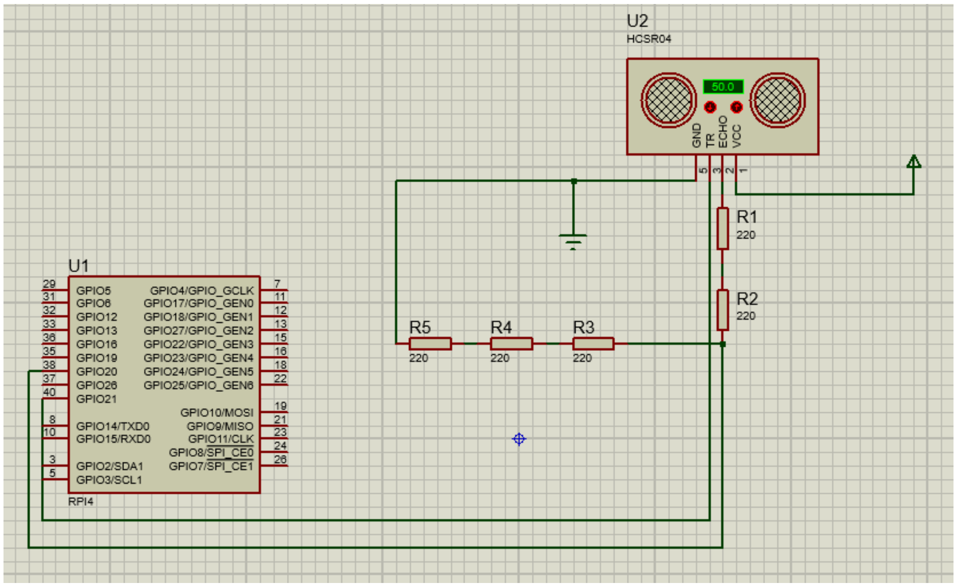
We will use the ultrasonic sensor to measure the distance by utilizing the difference between the start and end times in the distance formula, distance = speed * time. We used the speed of sound which is 17150 cm/s. The calculated distance was divided by 2 because the sound wave travels twice the distance (to move away and return). At last, we were able to see the distances on the Raspberry Pi code interface when we executed the code on the device. We were able to conduct our experiment without any significant issues.

2. Components:

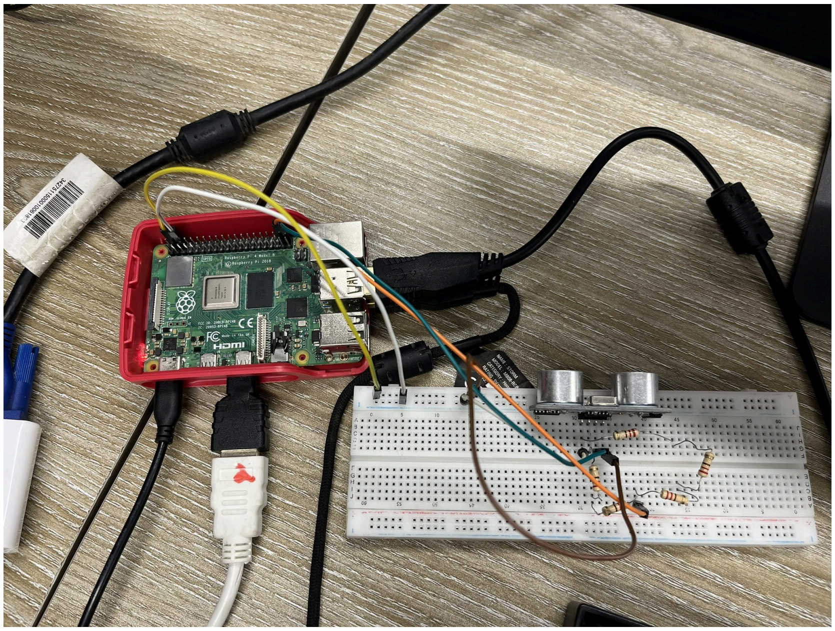
- Raspberry Pi 4
- Breadboard
- Ultrasonic sensor
- 220-ohm resistor

- Jumper wires (male-to-male, male-to-female, female-to-female)
- Micro SD Card
- Keyboard, Mouse and Monitor

3. Circuit Diagram:



4. Circuit Setup:



5. Explanation:

The task involved connecting the sonar sensor's TRIG and ECHO pins in order to measure distance. The sonar sensor's TRIG pin connects to the Raspberry Pi's GPIO 21 (actual PIN 40). To limit the current or voltage, the sonar sensor's ECHO pin connects to GPIO 20 (the physical PIN 38) of the Raspberry Pi using two resistances. The Raspberry Pi's 5V and GND pins are linked to the sonar sensor's 5V and GND pins, respectively. The resistors were connected in parallel with a 2:3 ratio. Five 220 ohm resistors were used in total to establish such a connection. We were able to operate the circuit by writing code once the pins were connected and joined to ground and voltage. We calculated the time taken for the sound wave to be received from the time it was emitted, and then used the formula distance = speed * time. Speed of sound, which is 17150 cm/s was used in the calculation. The calculated distance was divided by 2 because the sound wave travels twice the distance (to move away and return). This allowed us to successfully measure the distance between the sonar sensor and an object in centimeters.

6. 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()
```

7. Question- Answer:

1) Why are the resistors used?

   Answer:

The idea behind how ultrasonic sensors operate is the exchange of sound waves. The ultrasonic sensor's ECHO pin might be operating at a voltage greater than what the Raspberry Pi's GPIO pins can endure. In order to guarantee reliability and security, the voltage level from the ECHO pin must be lowered to an appropriate level that the Raspberry Pi can handle using a voltage divider that uses resistors. Resistors can be used in series with the power input to limit the current flow to a safe level if the ultrasonic sensor needs a current-limited power supply. Its primary function is to shield the Raspberry Pi from any possible damage.