Lab Worksheet 2

CSE461: Introduction to Robotics
Department of Computer Science and Engineering

Lab 02: Introduction to Raspberry PI microcontroller and control an LED with an ultrasonic sensor using the board.

I. Topic Overview

The lab is designed to introduce the basics of a microcontroller using Raspberry PI development board along with Raspberry PI OS. The students will get introduced to the pins of the board, their functions and lastly, learn how to use the GPIO - General Purpose Input Output pins for interfacing simple I/O devices like LED and ultrasonic sensors with the board. We will also be discussing the process of measuring distance using an ultrasonic sensor with the help of a Raspberry Pi. The aim of this lab is to demonstrate how to use an ultrasonic sensor with Raspberry Pi to measure distance accurately.

II. Learning Outcome

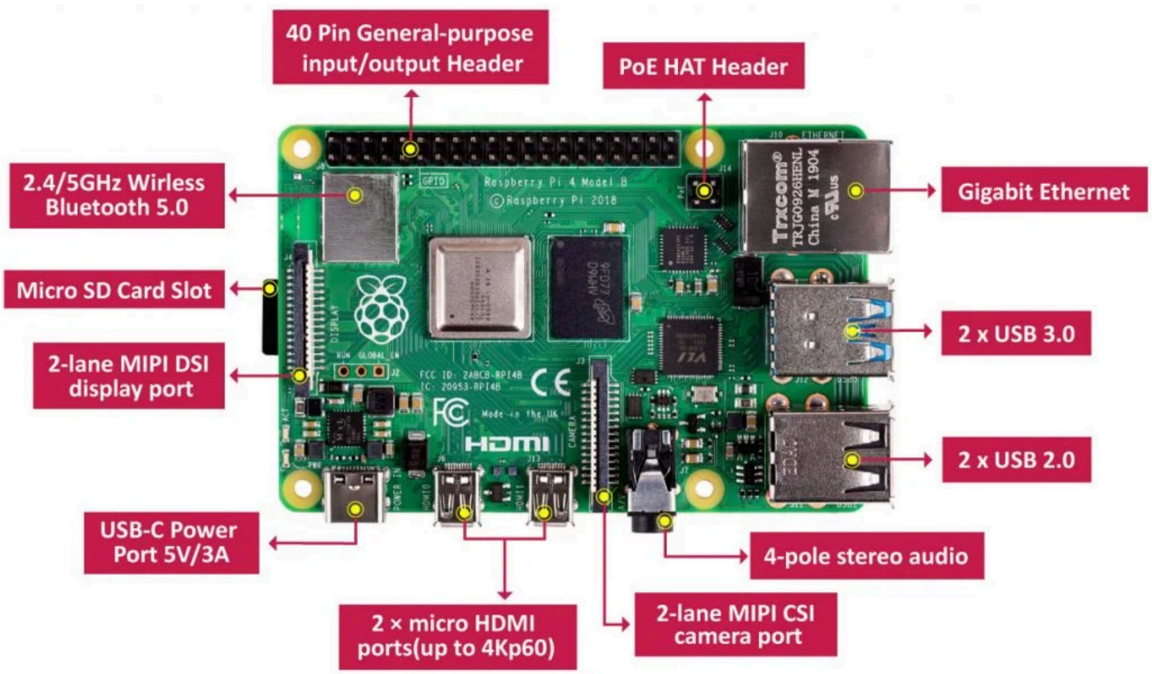
After this lab, students will be able to:

1. Know the various pins of the Raspberry PI board and their functions.
2. Build a simple circuit with the board and I/O devices.
3. Code to configure the GPIO pins to control different circuits for different usages.
4. Understand the basic principles of ultrasonic sensors and how they can be used to measure distance.
5. Learn how to install Python libraries on the Raspberry Pi and how to create and run Python programs using the terminal.

III. Materials

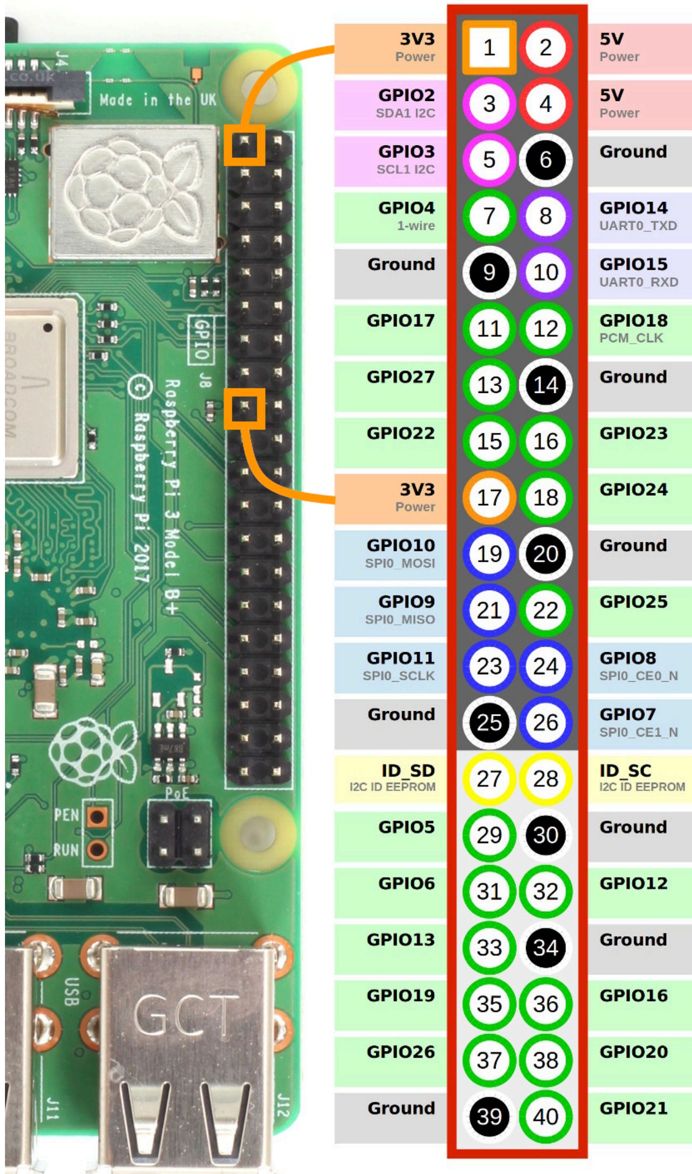
- Raspberry PI development board
- LED (Light Emitting Diode)
- Ultrasonic Sensor(HC-SR04)
- Resistor (appropriate value for current limiting, typically around 220 ohms)
- Jumper wires
- Breadboard
- USB pen drive containing OS

IV. Raspberry PI Overview



Raspberry Pi 4 Model B features a high-performance 64-bit quad-core processor, dual-display support at resolutions up to 4K via a pair of micro HDMI ports, hardware video decode at up to 4Kp60, up to 8GB of RAM, dual-band 2.4/5.0 GHz wireless LAN, Bluetooth 5.0, Gigabit Ethernet, USB 3.0, and PoE capability (via a separate PoE HAT add-on). For the end user, Raspberry Pi 4 Model B provides desktop performance comparable to entry-level x86 PC systems. Raspberry Pi OS (previously called Raspbian) is the recommended operating system for normal use on a Raspberry Pi.[Raspberry Pi Imager] is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card or a pendrive, ready to use with your Raspberry Pi. Then the board can be programmed using user-friendly IDEs, for example python programming can be done using Thonny IDE.

V. Raspberry PI Pins:



The Raspberry PI 4B has the following types of pins:

1. **Power Pins:**
    a. **5V and 3.3V:** Provide regulated voltage output.
    b. **GND:** Ground pins.

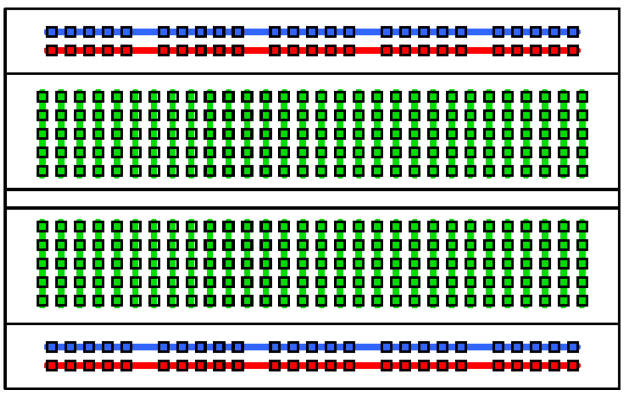
2. **GPIO Pins:** The general purpose input and output pins. Some of these GPIO pins can be used for following different communication protocols like UART, I2C and SPI to communicate with other devices. The GPIO pins work at a high voltage of 1.8V-3.3V maximum.

V. Raspberry PI OS Installation Guideline: Setting up Raspberry PI

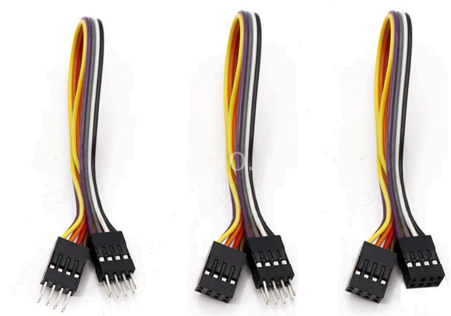
VII. Experiment 1: LED blinking.

1. **Description the components:**

a. Breadboard



b. Jumper wires

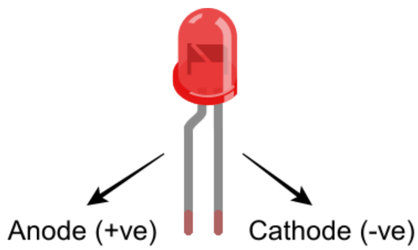


c. Resistors

4 Band Resistor Color Code				
Color	1. Band	2. Band	3. Band Multiplier	4. Band Tolerance
black	-	0	1	-
brown	1	1	10	+/- 1%
red	2	2	100	+/- 2%
orange	3	3	1'000	-
yellow	4	4	10'000	-
green	5	5	100'000	-
blue	6	6	1'000'000	-
purple	7	7	-	-
grey	8	8	-	-
white	9	9	-	-
gold	-	-	0.1	+/- 5%
silver	-	-	0.01	+/- 10%

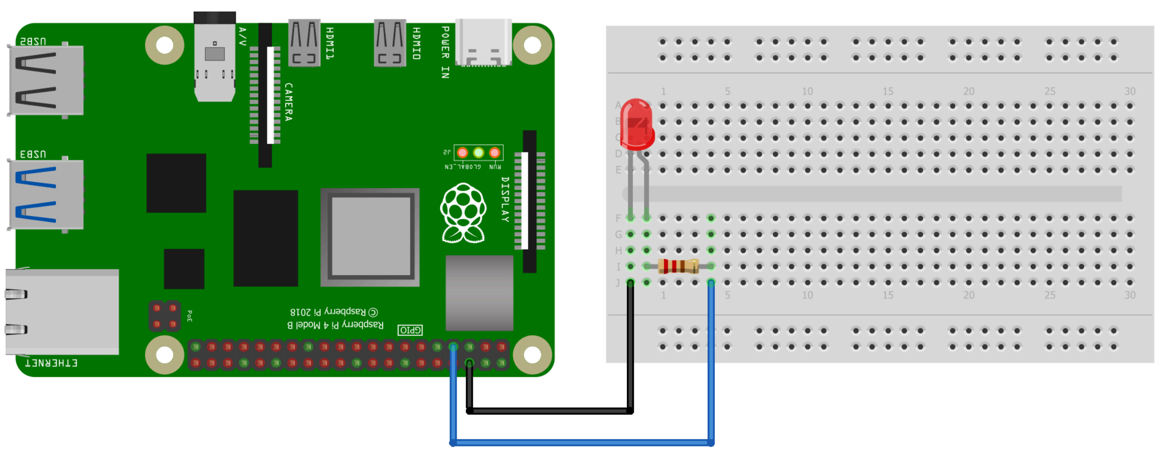
Table 1: 4 band resistor color codes

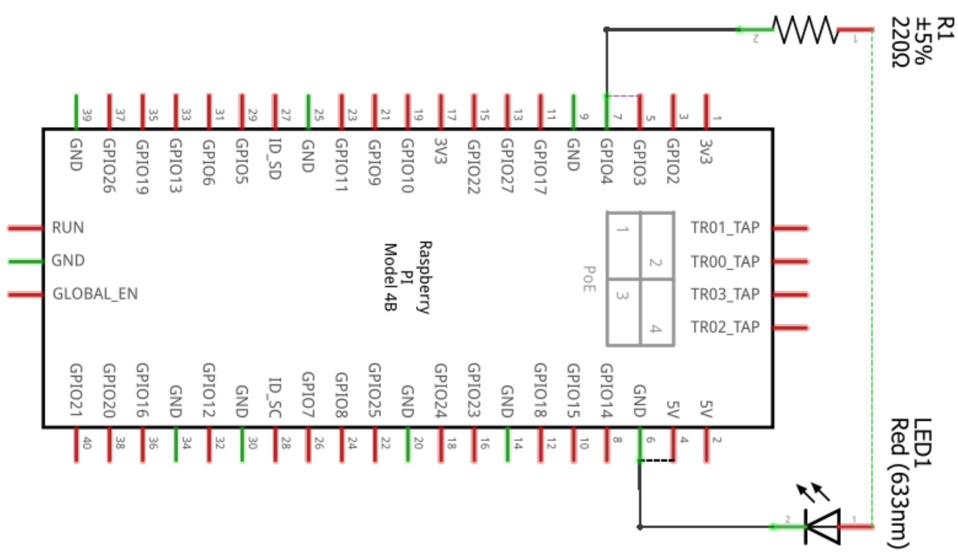
d. LED



2. **Setting up the circuit:**
    - Connect one end of a 220 ohm resistor to a GPIO pin of your Raspberry PI and connect the other end to the anode of the LED.
    - Connect the cathode of the LED to a ground pin of the board.

3. **Circuit diagram:**





4. **Code:** Open a .py file in a folder on the RPI OS. Open Thonny IDE and save the code in the previously created file. Then run the code.

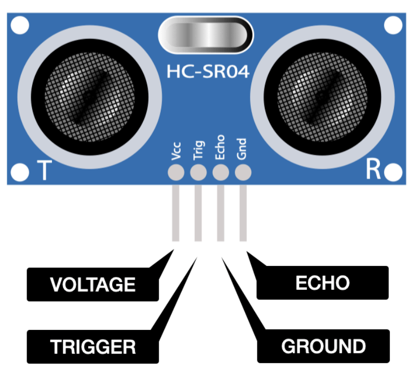
```
from gpiozero import LED
#imports LED functions from gpiozero library
import time
#imports time library

led = LED(4)
#declare the GPIO pin 4 for LED output and store it in led variable

while True:
#initiated an infinite while loop
    led.on()
    #turn on the led
    time.sleep(1)
    #delay for 1 second (increase the delay if necessary)
    led.off()
    #turn off the led
    time.sleep(1)
    #delay for 1 second (increase the delay if necessary)
```

VIII. Experiment 2: Measuring distance using ultrasonic sensor.

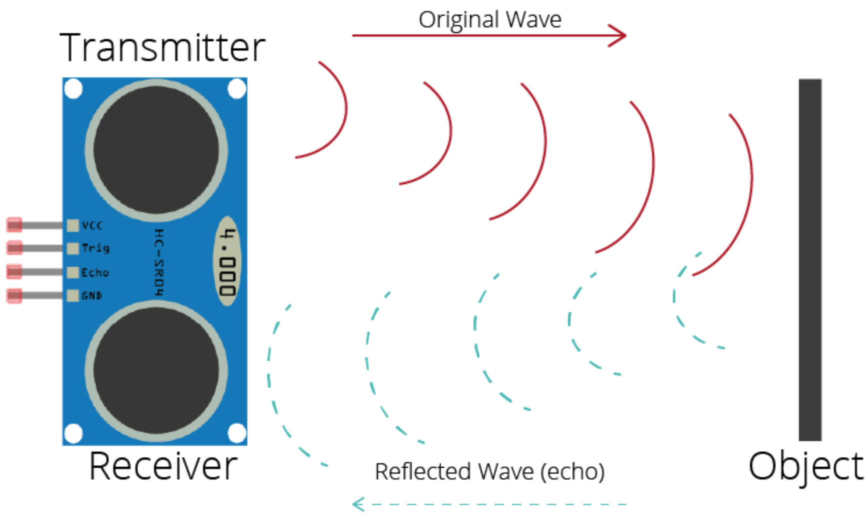
1. **Description the components:**
    a. **Ultrasonic Sensor:** Ultrasonic sensors are a common type of sensor that are used for measuring distances by sending out sound waves and measuring the time it takes for them to bounce back.

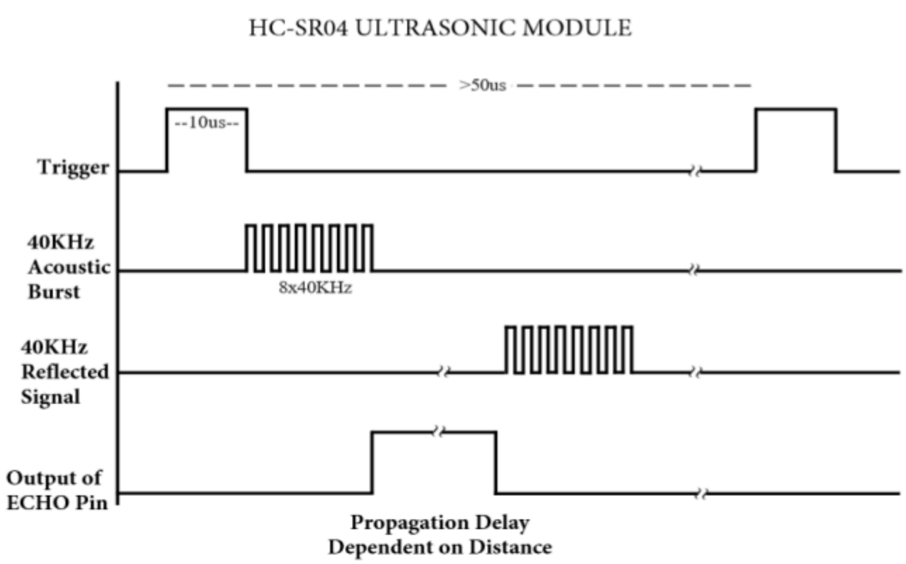


- **Pin1 (Vcc):** This pin provides a +5V power supply to the sensor.
- **Pin2 (Trigger):** This is an input pin, used to initialize measurement by transmitting ultrasonic waves by keeping this pin high for 10us.
- **Pin3 (Echo):** This is an output pin, which goes high for a specific time period and it will be equivalent to the duration of the time for the wave to return back to the sensor.

- **Pin4 (Ground):** This is a GND pin used to connect to the GND of the system.

Working mechanism of an ultrasonic sensor:





- A microcontroller is used for communication with an ultrasonic sensor.
- To begin measuring the distance, the microcontroller sends a trigger signal to the ultrasonic sensor. The duty cycle of this trigger signal is 10µS for the HC-SR04 ultrasonic sensor.
- When triggered, the ultrasonic sensor generates eight acoustic (ultrasonic) wave bursts and initiates a time counter.

- As soon as the reflected (echo) signal is received, the timer stops. The output of the ultrasonic sensor is a high pulse with the same duration as the time difference between transmitted ultrasonic bursts and the received echo signal.

2. **Setting up the circuit:**
    - Connect the VCC pin of the ultrasonic sensor to the 5V pin of the board.
    - Then the GND pin with the ground pin of the board.
    - Connect the TRIG (trigger pin) and the ECHO pin with GPIO pin 21 and 20.
    - The ECHO pin outputs 5V which will damage the GPIO pin of the Raspberry PI as it can only take 3.3V at maximum. So in order to convert it to a safer level we will use resistors using the voltage divider rule shown in the figure below. The 5V from the Echo pin will first need to be converted to 3.3V and then connect it to the GPIO pin. Then the voltage needs to be dropped to the ground. The ratio of the voltage division has to be 1:2 where we will be using one 220 ohm resistor connected to 5V and two 220 ohm resistors to GND.

Voltage Divider Formula:

The general formula for a voltage divider is:

$$V_{out} = V_{in} \times \frac{R_1}{R_1 + R_2}$$

where:

• V_{in} = 5V

• V_{out} = 3.3V

• R_1 is the resistor connected to 5V

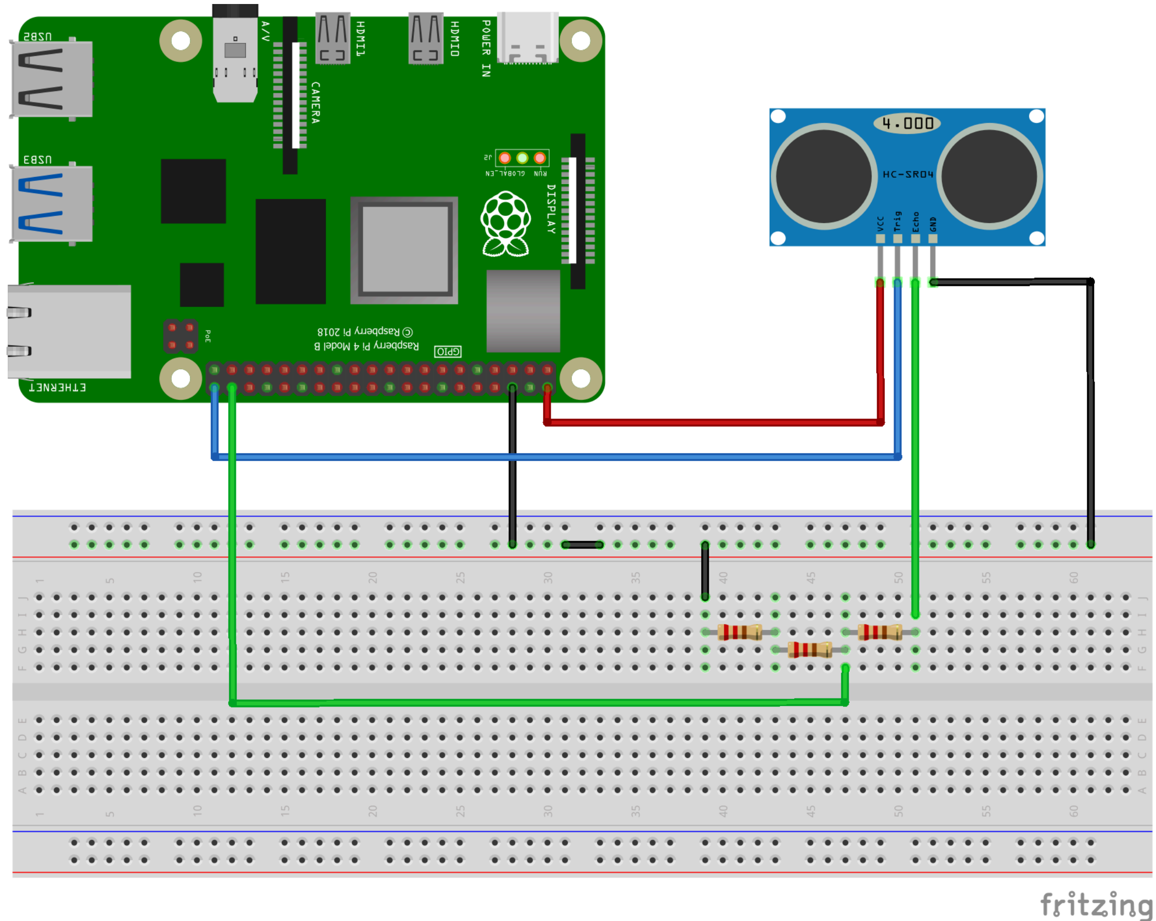
• R_2 is the resistor connected to GND

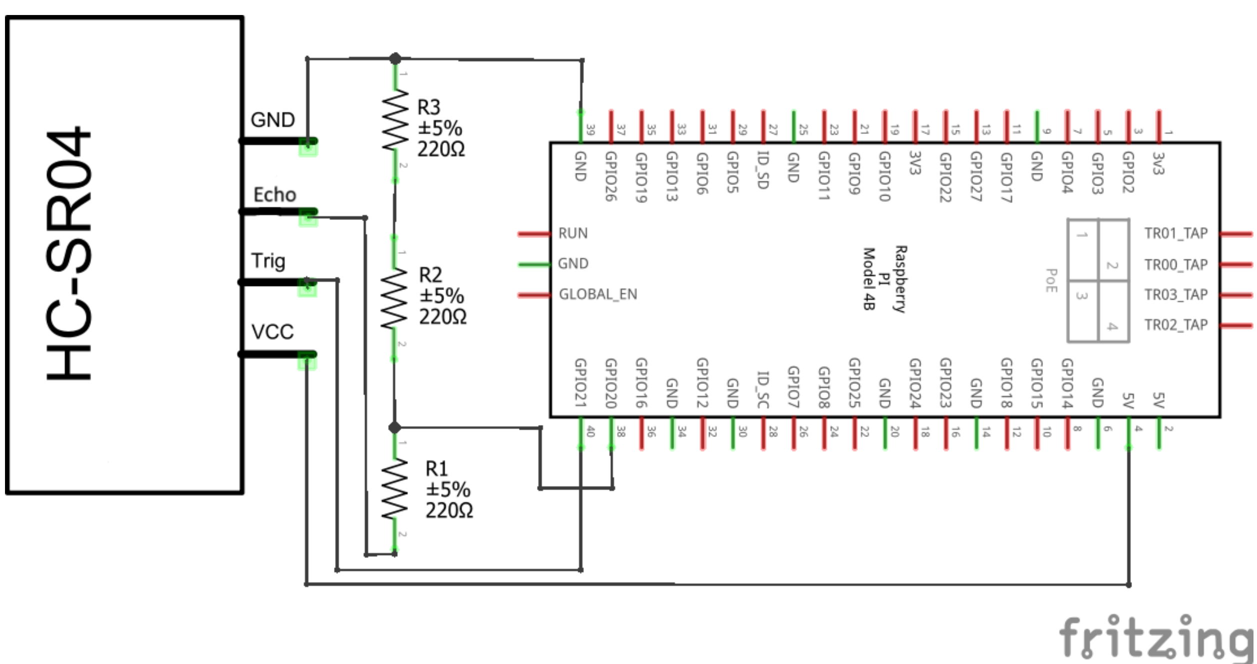
Using three 220 ohm resistors:
One resistor (220 ohm) as R_1
Two resistors (220 ohm each) in series as $R_4 = R_2 + R_3 =$ 220 ohm + 220 ohm = 440 ohm
Now, applying the voltage divider formula:

$$V_{out} = V_{in} \times \frac{R_1}{R_1 + R_4} = 5V \times \frac{220\Omega}{220\Omega + 440\Omega} \approx 3.33\text{V}$$

This is close to 3.3V, which is safe for the Raspberry Pi GPIO pins.

3. **Circuit diagram:**





4. **Code:** Run the code in Thonny IDE

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
TRIG = 21 #GPIO21
ECHO = 20 #GPIO20

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()
```

IX. Lab Task: Turn an LED on if the distance measured by the ultrasonic sensor is less than 5CM.

X. References:

1. https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf
2. https://www.raspberrypi.com/documentation/computers/getting-started.html