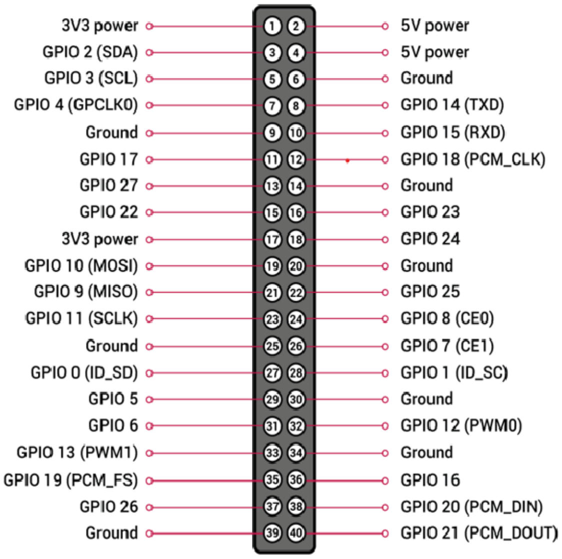
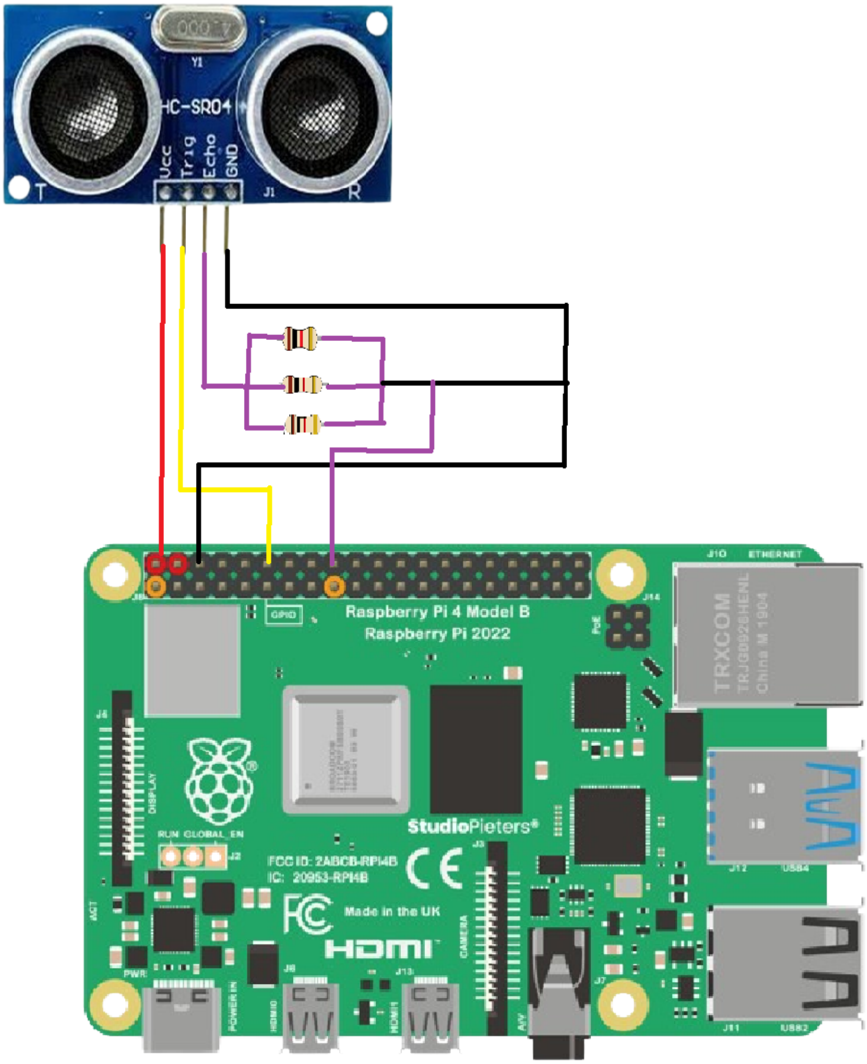
INTERFACING OF ULTRASONIC SENSOR WITH RASPBERRY PI

CIRCUIT DIAGRAM:





Code:

```
#Libraries
import RPi.GPIO as GPIO
import time
```

Step 1

```
#GPIO Mode (BOARD / BCM)
GPIO.setmode(GPIO.BCM)

#set GPIO Pins
GPIO_TRIGGER = 18
GPIO_ECHO = 24

#set GPIO direction (IN / OUT)
GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
GPIO.setup(GPIO_ECHO, GPIO.IN)
```

Step 2

```
def distance():
    # set Trigger to HIGH
    GPIO.output(GPIO_TRIGGER, True)

    # set Trigger after 0.01ms to LOW
    time.sleep(0.00001)
    GPIO.output(GPIO_TRIGGER, False)

    StartTime = time.time()
    StopTime = time.time()

    # save StartTime
    while GPIO.input(GPIO_ECHO) == 0:
        StartTime = time.time()

    # save time of arrival
    while GPIO.input(GPIO_ECHO) == 1:
        StopTime = time.time()

    # time difference between start and arrival
    TimeElapsed = StopTime - StartTime
    # multiply with the sonic speed (34300 cm/s)
    # and divide by 2, because there and back
    distance = (TimeElapsed * 34300) / 2

    return distance
```

Step 3

```
if __name__ == '__main__':
    try:
        while True:
            dist = distance()
            print ("Measured Distance = %.1f cm" % dist)
            time.sleep(1)
```

Step 4

```
        # Reset by pressing CTRL + C
    except KeyboardInterrupt:
        print("Measurement stopped by User")
        GPIO.cleanup()
```

Step 5

Explanation:

Step 1

Import Libraries: The code begins by importing the necessary libraries, including RPi.GPIO for GPIO control and time for timing operations.

Step 2

GPIO Configuration:

- It sets the GPIO mode to BCM (Broadcom SOC channel numbering).
- Defines the GPIO pins for the ultrasonic sensor's trigger (GPIO_TRIGGER) and echo (GPIO_ECHO).
- Configures the GPIO pins as either input or output.

Step 3

Distance Measurement Function (distance()):

- The distance() function calculates the distance to an object using the time taken for an ultrasonic pulse to travel to the object and back.
- It triggers the sensor by setting the trigger pin to HIGH for a short duration (10 microseconds) and then LOW.
- Measures the time it takes for the echo signal to transition from LOW to HIGH (start time) and then from HIGH to LOW (stop time).
- Calculates the time elapsed and converts it to distance using the speed of sound (34300 cm/s) and dividing by 2 (since the signal travels to the object and back).

Step 4

Main Loop:

- The script enters an infinite loop to continuously measure distances.
- Calls the distance() function to get the distance in centimeters.
- Prints the measured distance.
- Sleeps for 1 second before repeating the measurement.

Step 5

Exception Handling:

- The script handles the Keyboard Interrupt exception (Ctrl+C) to stop the measurement and clean up the GPIO pins.