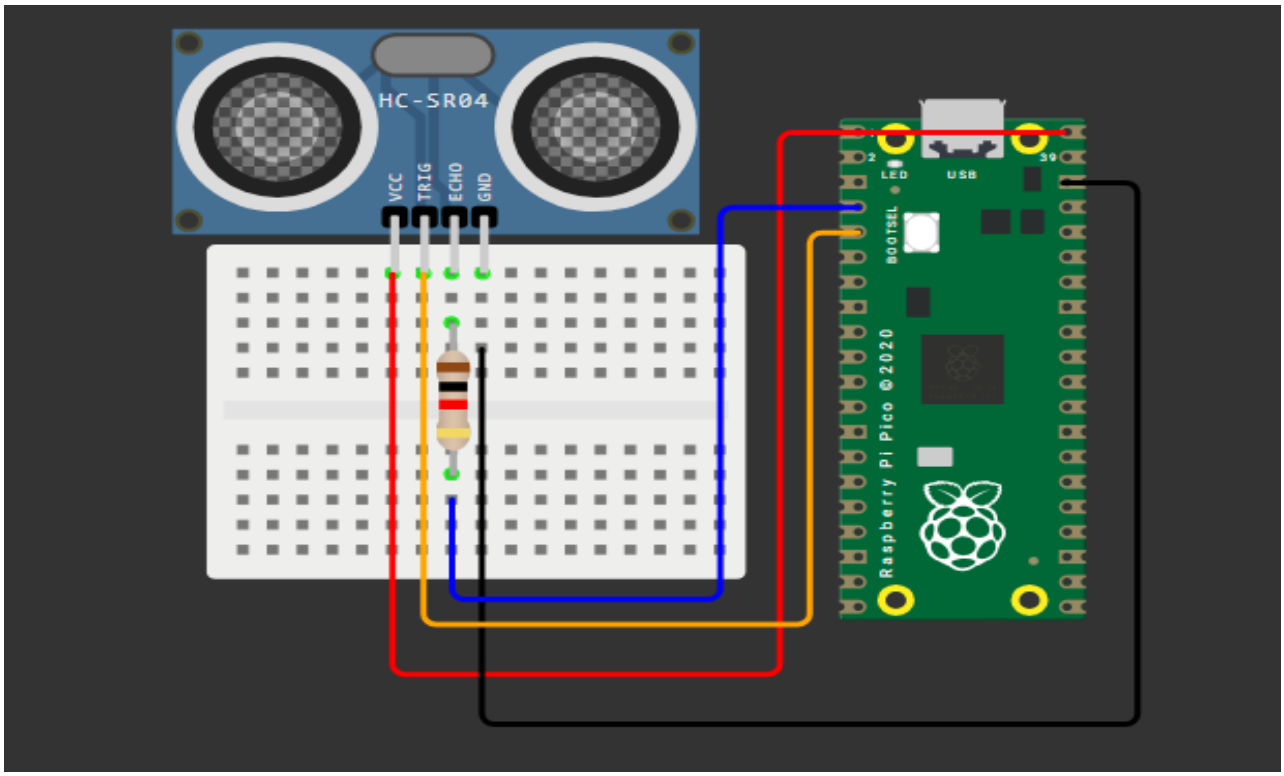
PRACTICAL 9

Interfacing Raspberry Pi Pico 2 with Ultrasonic Sensor (HC-SR04)

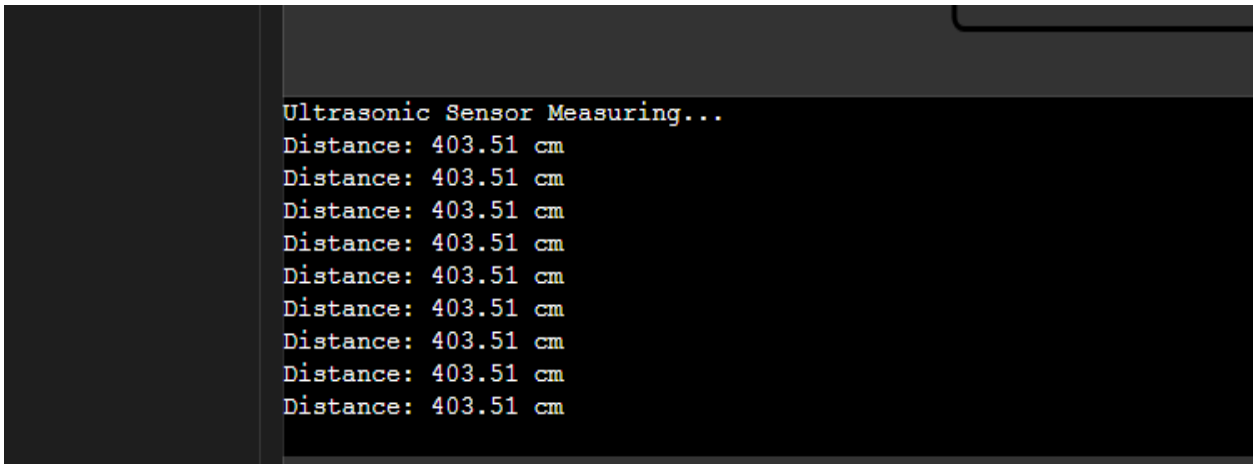


PROGRAM:

```
from machine import Pin, time_pulse_us
from time import sleep

# Define Pins
TRIG = Pin(3, Pin.OUT)
ECHO = Pin(2, Pin.IN)

def get_distance():
    # Send a 10µs pulse to trigger
    TRIG.low()
    sleep(0.002)
```

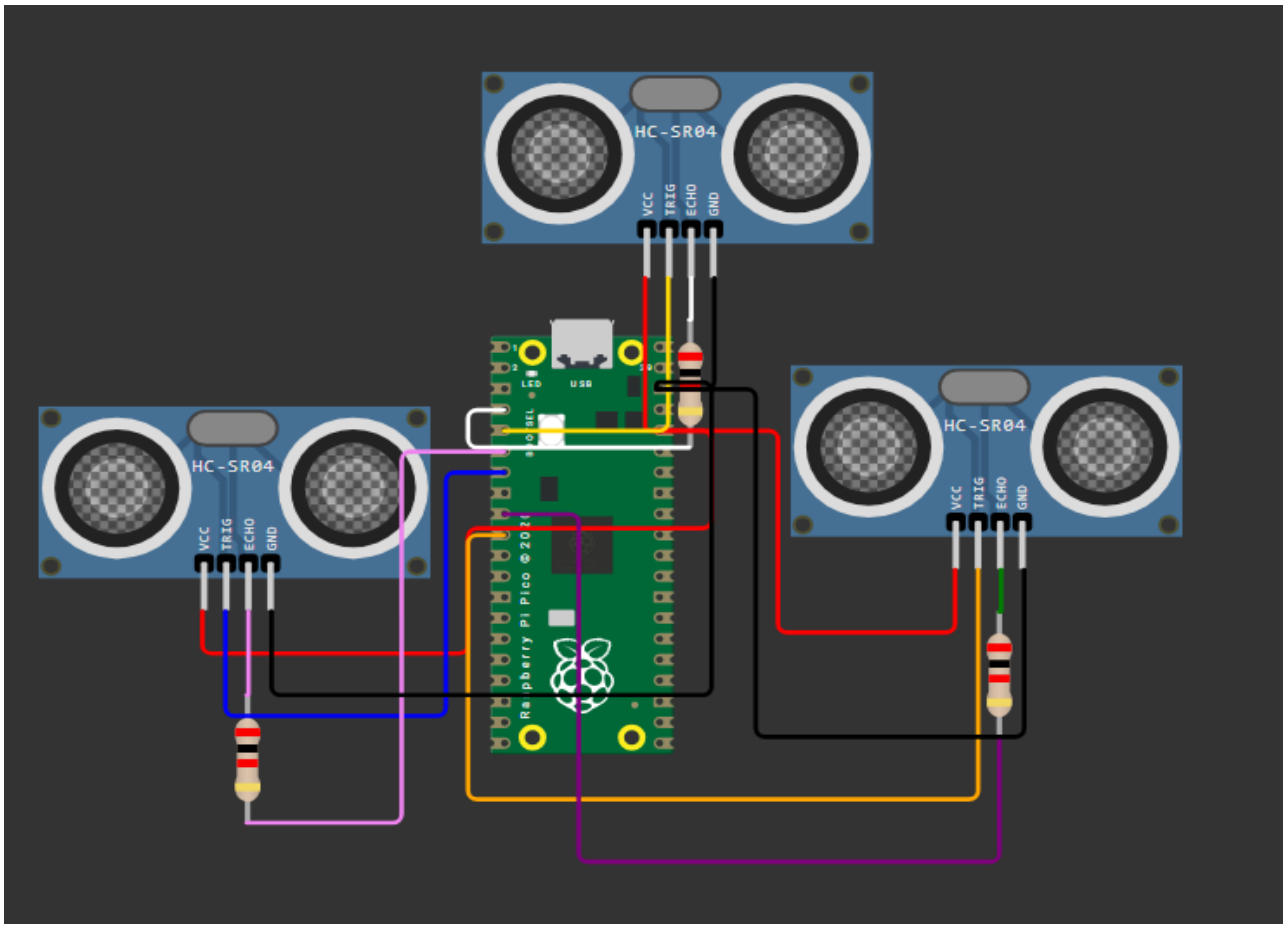
```
    TRIG.high()
    sleep(0.00001)
    TRIG.low()

    # Measure the duration of echo pulse
    duration = time_pulse_us(ECHO, 1, 30000)  # timeout 30ms

    # Calculate distance in cm (speed of sound = 343m/s)
    distance = (duration * 0.0343) / 2
    return distance

print("Ultrasonic Sensor Measuring...")

while True:
    dist = get_distance()
    print("Distance: {:.2f} cm".format(dist))
    sleep(0.5)
```

OUTPUT:



ASSIGNMENT 9

Create a parking assistant system using multiple sensors.



PROGRAM:

```
from machine import Pin, time_pulse_us
from time import sleep


# Define Sensors
sensors = {
    "Front": {"trig": Pin(3, Pin.OUT), "echo": Pin(2, Pin.IN), "led": Pin(10, Pin.OUT)},
    "Left":  {"trig": Pin(5, Pin.OUT), "echo": Pin(4, Pin.IN), "led": Pin(11, Pin.OUT)},
    "Right": {"trig": Pin(7, Pin.OUT), "echo": Pin(6, Pin.IN), "led": Pin(12, Pin.OUT)},
```

```
}

buzzer = Pin(15, Pin.OUT)

def get_distance(trig, echo):
    trig.low()
    sleep(0.002)
    trig.high()
    sleep(0.00001)
    trig.low()
    duration = time_pulse_us(echo, 1, 30000)  # Timeout 30ms
    distance = (duration * 0.0343) / 2
    return distance

print("Parking Assistant System Active...\n")

while True:
    warning = False

    for name, s in sensors.items():
        dist = get_distance(s["trig"], s["echo"])
        print(f"{name} Distance: {dist:.1f} cm")

        if dist < 15:  # Obstacle within 15 cm
            s["led"].value(1)
            warning = True
        else:
            s["led"].value(0)

    # Activate buzzer if any obstacle is too close
```

```
    buzzer.value(1 if warning else 0)


    print("----------------------------")
    sleep(0.5)
```

OUTPUT: