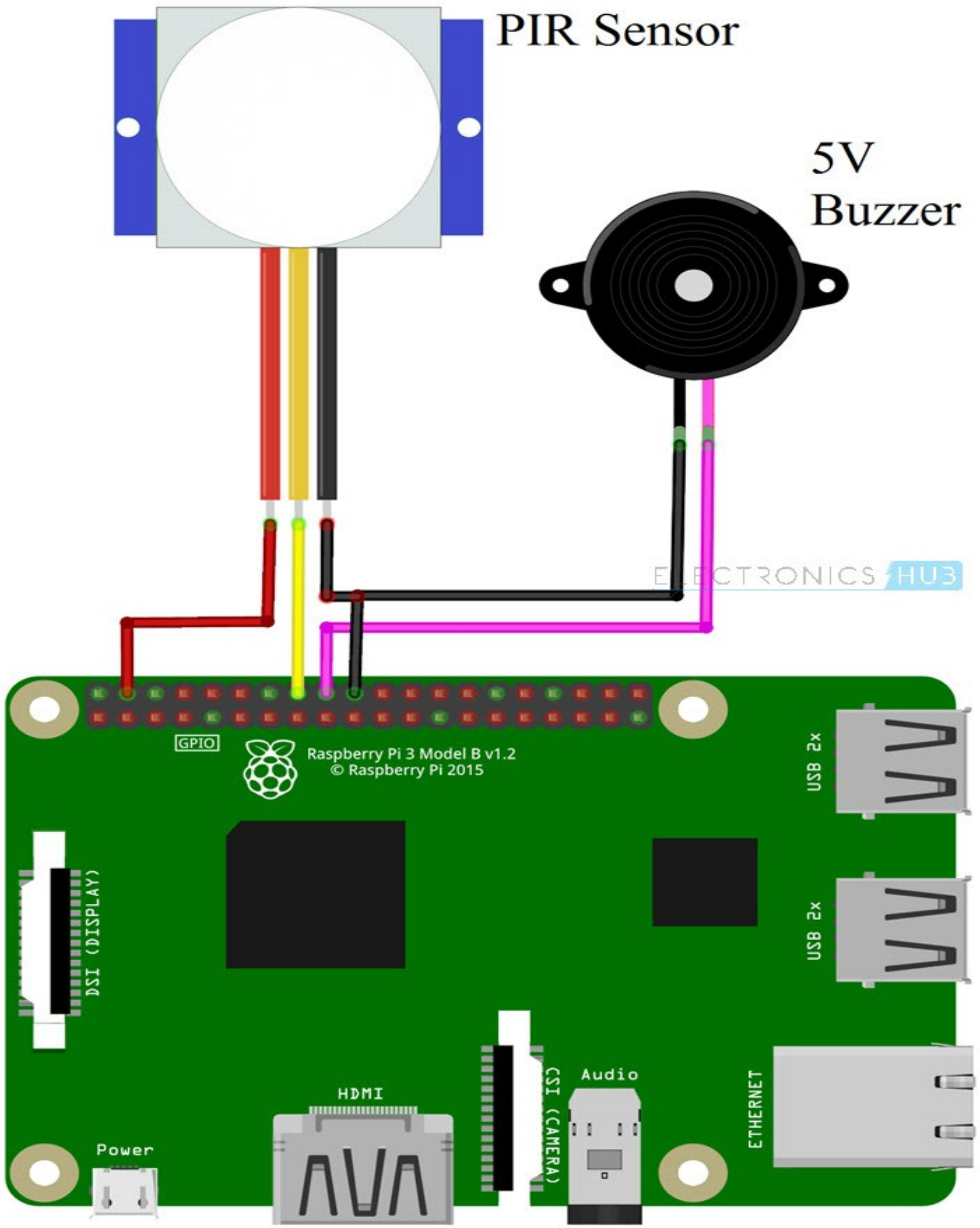
Experiment-8,9

1. Interface Ultrasonic sensor with Raspberry Pi
   (Integrate ultrasonic sensor to LED output)
2. Identify the MQ2 sensor output through the buzzer

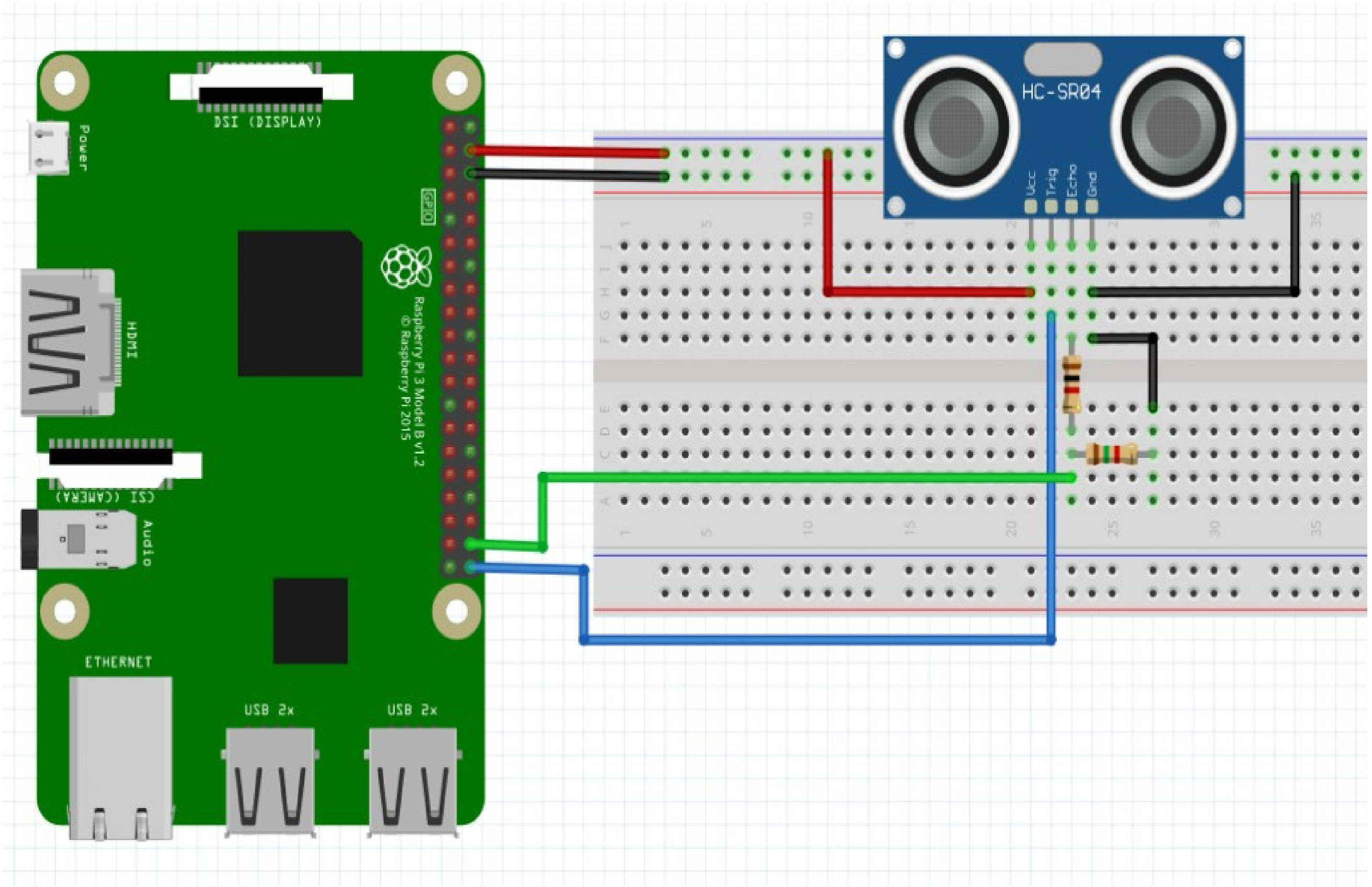
Interfacing Buzzer with Raspberry Pi 4

```
import RPi.GPIO as GPIO

import time

# Set GPIO mode

GPIO.setmode(GPIO.BOARD)

# Set the pin numbers for PIR sensor and buzzer

pir_pin = 17

buzzer_pin = 18

# Setup PIR pin as input

GPIO.setup(pir_pin, GPIO.IN)

# Setup buzzer pin as output

GPIO.setup(buzzer_pin, GPIO.OUT)

try:

    print("PIR Module Test (CTRL+C to exit)")

    time.sleep(2)  # Allow sensor to settle

    print("Ready")

    while True:

        if GPIO.input(pir_pin):  # Check if motion is detected

            print("Motion detected!")

            GPIO.output(buzzer_pin, GPIO.HIGH)  # Activate the buzzer

            time.sleep(1)  # Sound the buzzer for 1 second

            GPIO.output(buzzer_pin, GPIO.LOW)   # Deactivate the buzzer

        else:

            print("No motion detected")

        time.sleep(0.1)  # Check for motion every 0.1 second

except KeyboardInterrupt:

    pass

GPIO.cleanup()
```



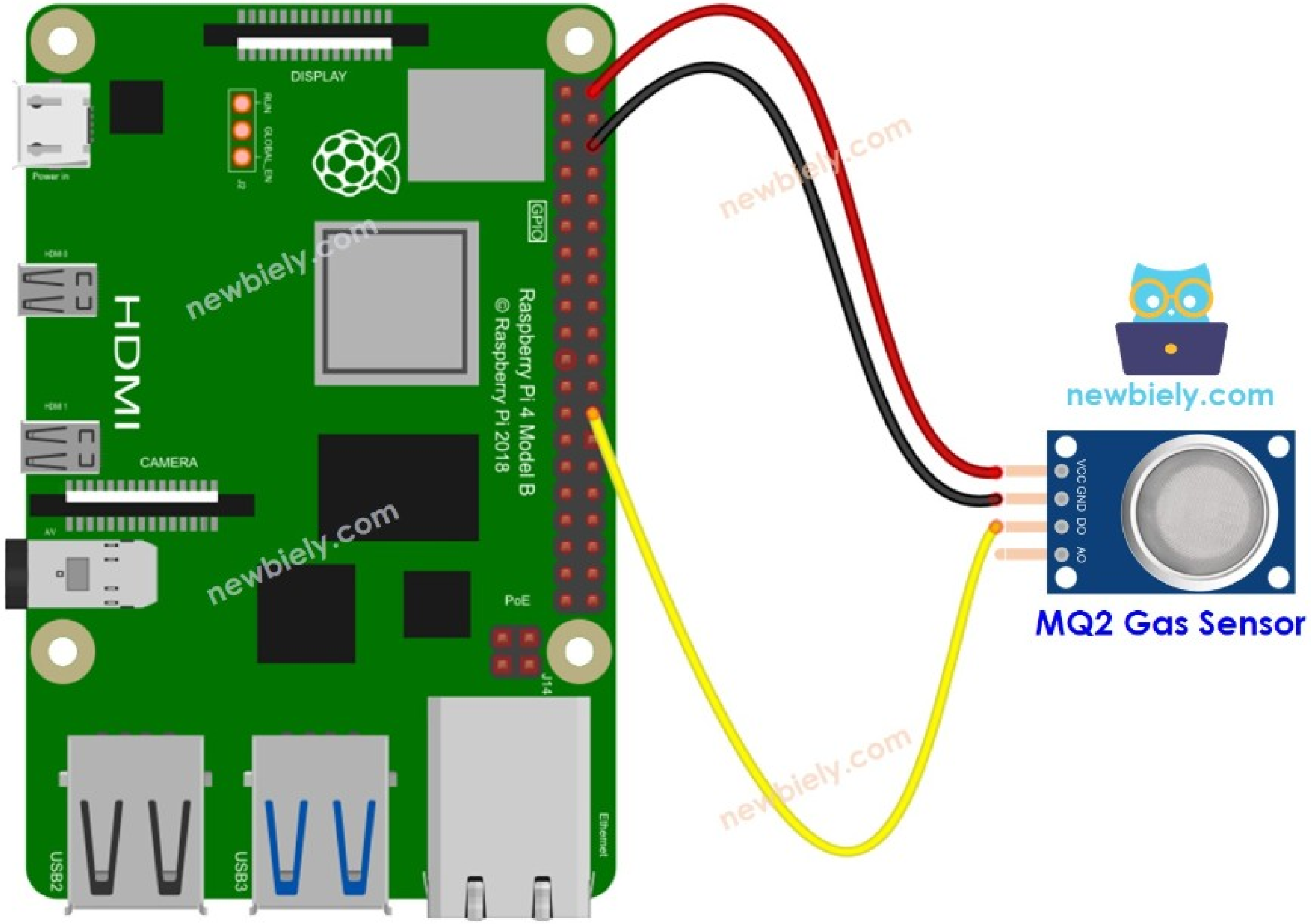
Interfacing Ultrasonic Sensor with Raspberry Pi 4

```
import RPi.GPIO as GPIO

import time

# Set GPIO mode

GPIO.setmode(GPIO.BOARD)

# Set the pin numbers for the ultrasonic sensor

trigger_pin = 23

echo_pin = 24

# Setup ultrasonic sensor pins

GPIO.setup(trigger_pin, GPIO.OUT)

GPIO.setup(echo_pin, GPIO.IN)

try:

    while True:

        # Send a pulse to trigger the ultrasonic sensor

        GPIO.output(trigger_pin, GPIO.HIGH)

        time.sleep(0.00001)

        GPIO.output(trigger_pin, GPIO.LOW)

        # Wait for the echo pin to go high

        while GPIO.input(echo_pin) == 0:

            pulse_start_time = time.time()

        # Wait for the echo pin to go low

        while GPIO.input(echo_pin) == 1:

            pulse_end_time = time.time()

        # Calculate pulse duration

        pulse_duration = pulse_end_time - pulse_start_time

        # Calculate distance in centimeters

        distance = pulse_duration * 17150

        distance = round(distance, 2)

 print("Distance:", distance, "cm")
        time.sleep(1)

except KeyboardInterrupt:
    pass

GPIO.cleanup()
```



Interfacing Gas Sensor (MQ2) with Raspberry Pi 4

```
import RPi.GPIO as GPIO
import time
# Set up the GPIO mode
GPIO.setmode(GPIO.BCM)
# Set up the GPIO pin for reading the DO output
DO_PIN = 7  # Replace with the actual GPIO pin n
GPIO.setup(DO_PIN, GPIO.IN)
try:
    while True:
        # Read the state of the DO pin
        gas_present = GPIO.input(DO_PIN)
        # Determine if gas is present or not
        if gas_present == GPIO.LOW:
            gas_state = "Gas Present"
        else:
            gas_state = "No Gas"
        # Print the gas state
        print(f"Gas State: {gas_state}")
        time.sleep(0.5)  # Wait for a short period before reading again
except KeyboardInterrupt:
    print("Gas detection stopped by user")

finally:
    # Clean up GPIO settings
    GPIO.cleanup()
```


DISPLAY
RUN
GLOBAL_EN
J2
Power in
GPIO
HDMI
HDMI 0
HDMI 1
Raspberry Pi 4 Model B
© Raspberry Pi 2018
CAMERA
AV
PoE
J14
USB2
USB3
Ethernet
newbiely.com
VCC
GND
DO
AO
MQ2 Gas Sensor