code: main.py

main.py diagram.json Library Manager

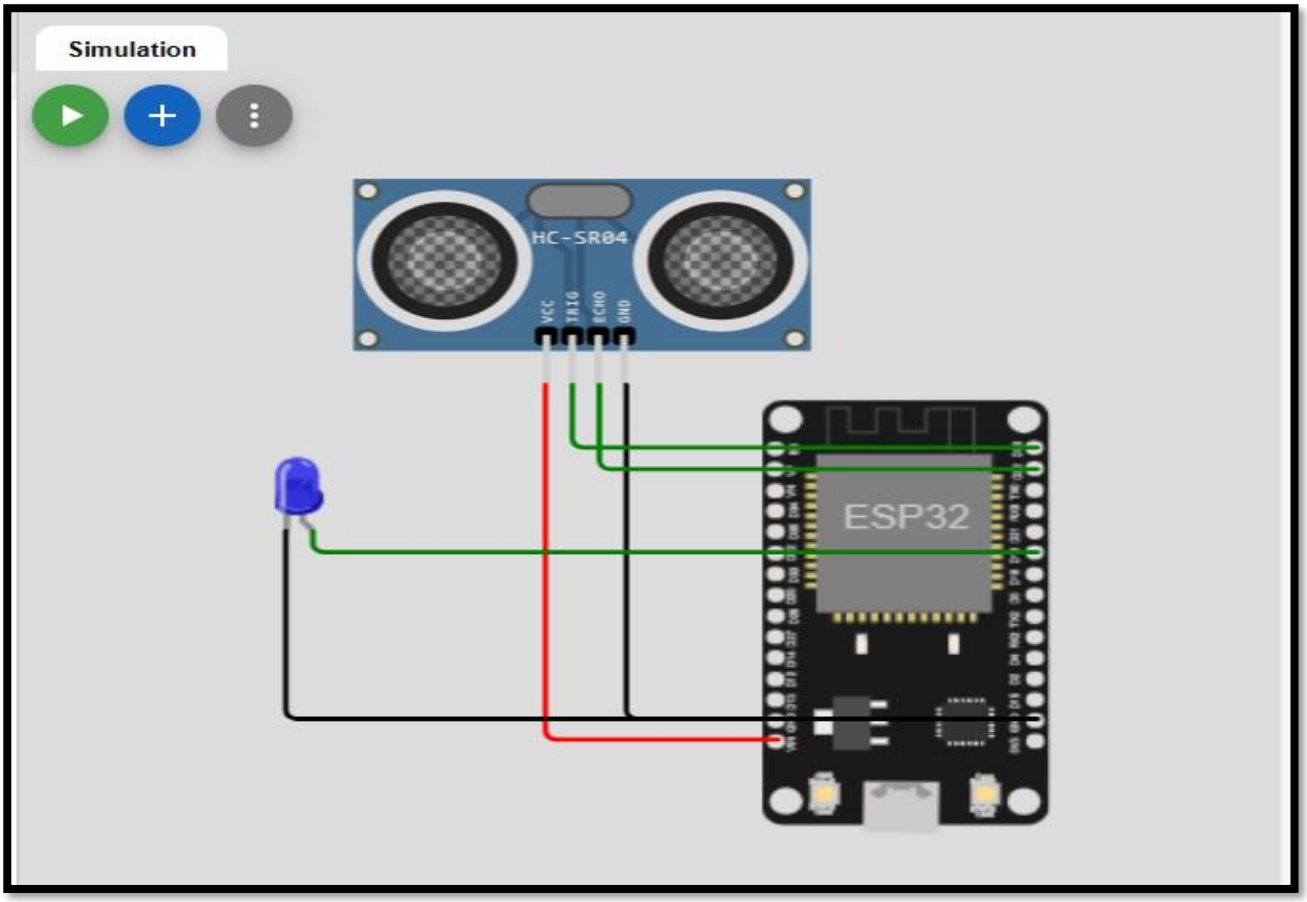
```python
import machine
import time

# Pin assignments for the ultrasonic sensor
TRIGGER_PIN = 23  # GPIO23 for trigger
ECHO_PIN = 22     # GPIO22 for echo

# Pin assignment for the LED
LEAK_LED_PIN = 19  # GPIO19 for the LED

# Set the pin modes
trigger = machine.Pin(TRIGGER_PIN, machine.Pin.OUT)
echo = machine.Pin(ECHO_PIN, machine.Pin.IN)
leak_led = machine.Pin(LEAK_LED_PIN, machine.Pin.OUT)

# Function to measure distance using the ultrasonic sensor
def measure_distance():
    # Generate a short trigger pulse
    trigger.value(0)
    time.sleep_us(5)
    trigger.value(1)
    time.sleep_us(10)
    trigger.value(0)

    # Measure the echo pulse duration to calculate distance
    pulse_start = pulse_end = 0
    while echo.value() == 0:
        pulse_start = time.ticks_us()
    while echo.value() == 1:
        pulse_end = time.ticks_us()

    pulse_duration = pulse_end - pulse_start

    # Calculate distance in centimeters (assuming the speed of sound is 343 m/s)
    distance = (pulse_duration * 0.0343) / 2  # Divide by 2 for one-way travel

    return distance

# Function to check for a water leak
def check_for_leak():
    # Measure the distance from the ultrasonic sensor
    distance = measure_distance()

    # Set the threshold distance for detecting a leak (adjust as needed)
    threshold_distance = 10  # Adjust this value based on your tank setup

    if distance < threshold_distance:
        # If the distance is less than the threshold, a leak is detected
        return True
    else:
        return False

# Main loop
while True:
    if check_for_leak():
        # Blink the LED to indicate a leak
        leak_led.value(1)  # LED ON
        time.sleep(0.5)
        leak_led.value(0)  # LED OFF
        time.sleep(0.5)
    else:
        leak_led.value(0)  # LED OFF

    time.sleep(1)  # Delay between measurements

```

S**diagram.json**

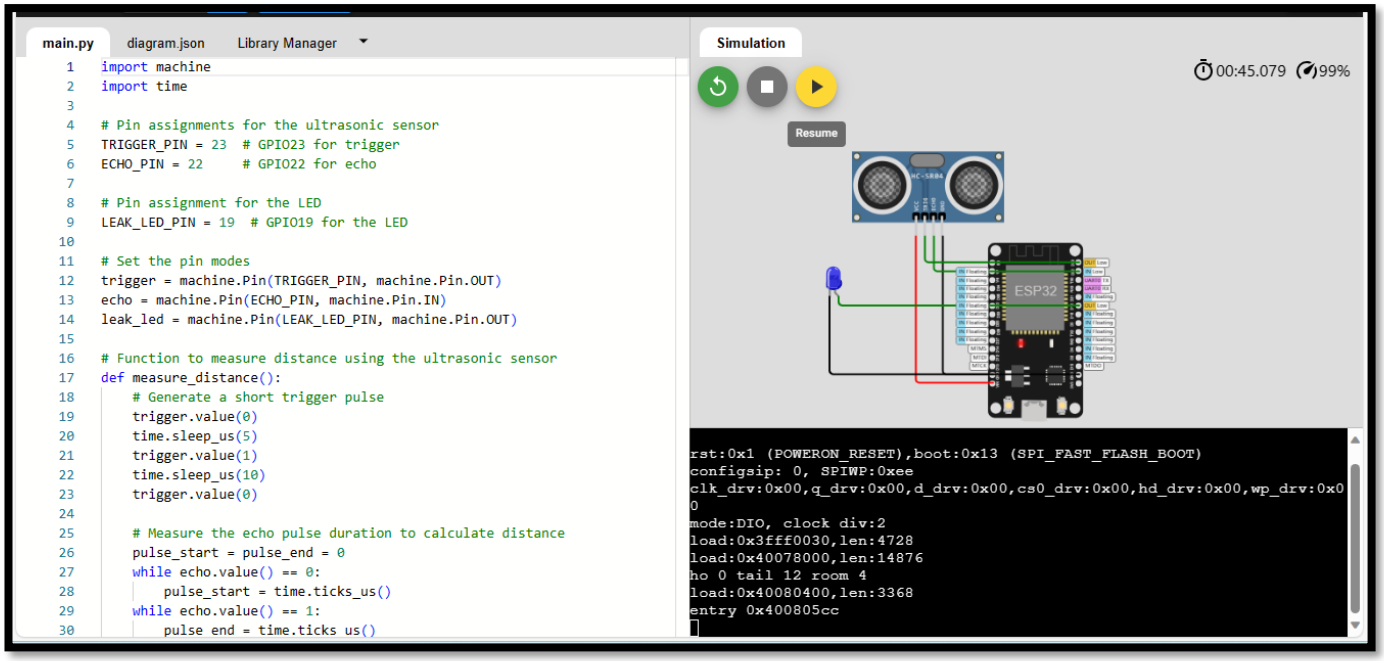
main.py | diagram.json | Library Manager

```
{
  "version": 1,
  "author": "Uri Shaked",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -14.5, "left": 81.4, "attrs": {} },
    { "type": "wokwi-led", "id": "led2", "top": 6, "left": -111.4, "attrs": { "color": "blue" } },
    { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -113.7, "left": -71.3, "attrs": {} }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
    [ "ultrasonic1:GND", "esp:GND.2", "black", [ "v0" ] ],
    [ "ultrasonic1:ECHO", "esp:D22", "green", [ "v0" ] ],
    [ "ultrasonic1:TRIG", "esp:D23", "green", [ "v0" ] ],
    [ "ultrasonic1:VCC", "esp:VIN", "red", [ "v0" ] ],
    [ "led2:A", "esp:D19", "green", [ "v0" ] ],
    [ "led2:C", "esp:GND.1", "black", [ "v0" ] ]
  ],
  "dependencies": {}
}
```

CIRCUIT :-



STIMULATION OUTPUT:-




```
import machine
import time

# Pin assignments for the ultrasonic sensor
TRIGGER_PIN = 23  # GPIO23 for trigger
ECHO_PIN = 22     # GPIO22 for echo

# Pin assignment for the LED
LEAK_LED_PIN = 19  # GPIO19 for the LED

# Set the pin modes
trigger = machine.Pin(TRIGGER_PIN, machine.Pin.OUT)
echo = machine.Pin(ECHO_PIN, machine.Pin.IN)
leak_led = machine.Pin(LEAK_LED_PIN, machine.Pin.OUT)

# Function to measure distance using the ultrasonic sensor
def measure_distance():
    # Generate a short trigger pulse
    trigger.value(0)
    time.sleep_us(5)
    trigger.value(1)
    time.sleep_us(10)
    trigger.value(0)

    # Measure the echo pulse duration to calculate distance
    pulse_start = pulse_end = 0
    while echo.value() == 0:
        pulse_start = time.ticks_us()
    while echo.value() == 1:
        pulse_end = time.ticks_us()
```