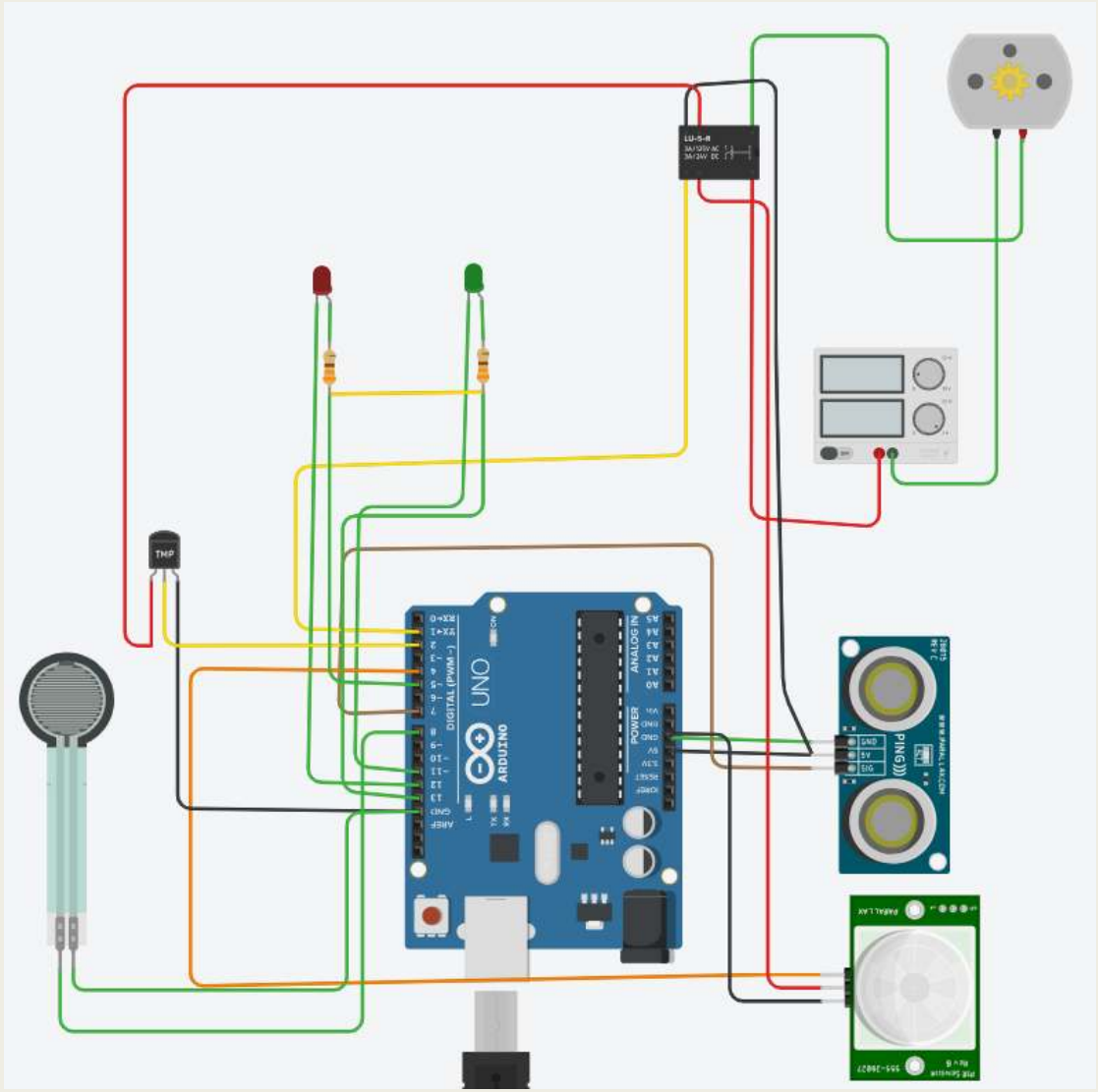
Smart water fountains

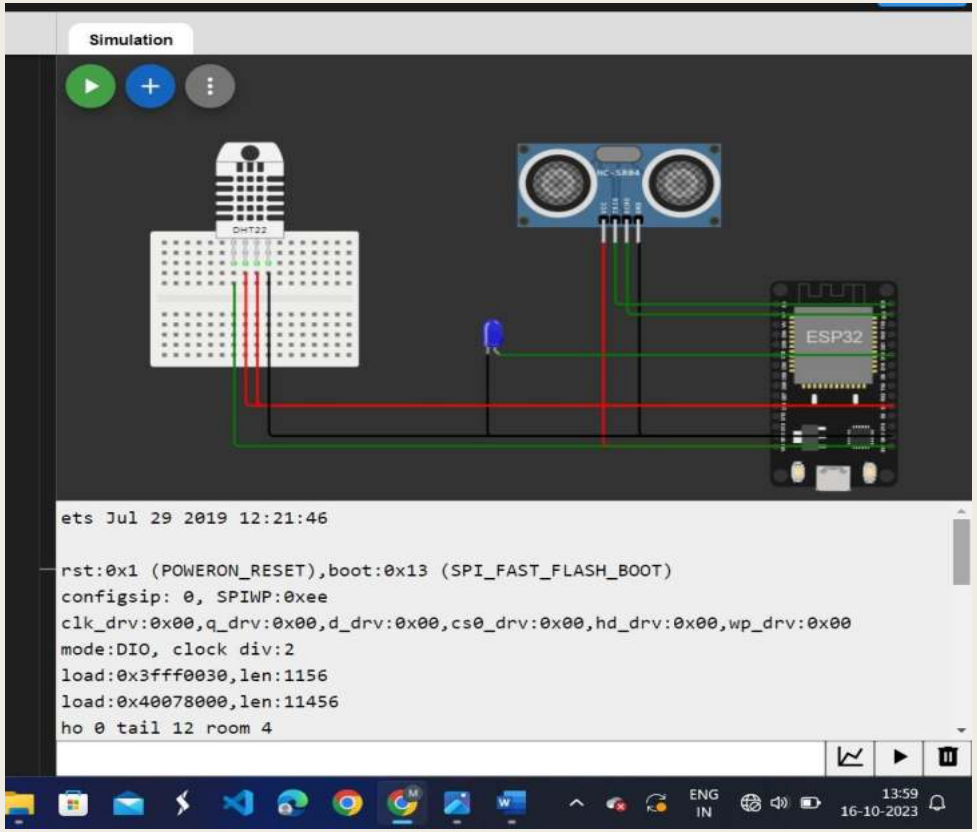
SMART WATER FOUNTAIN

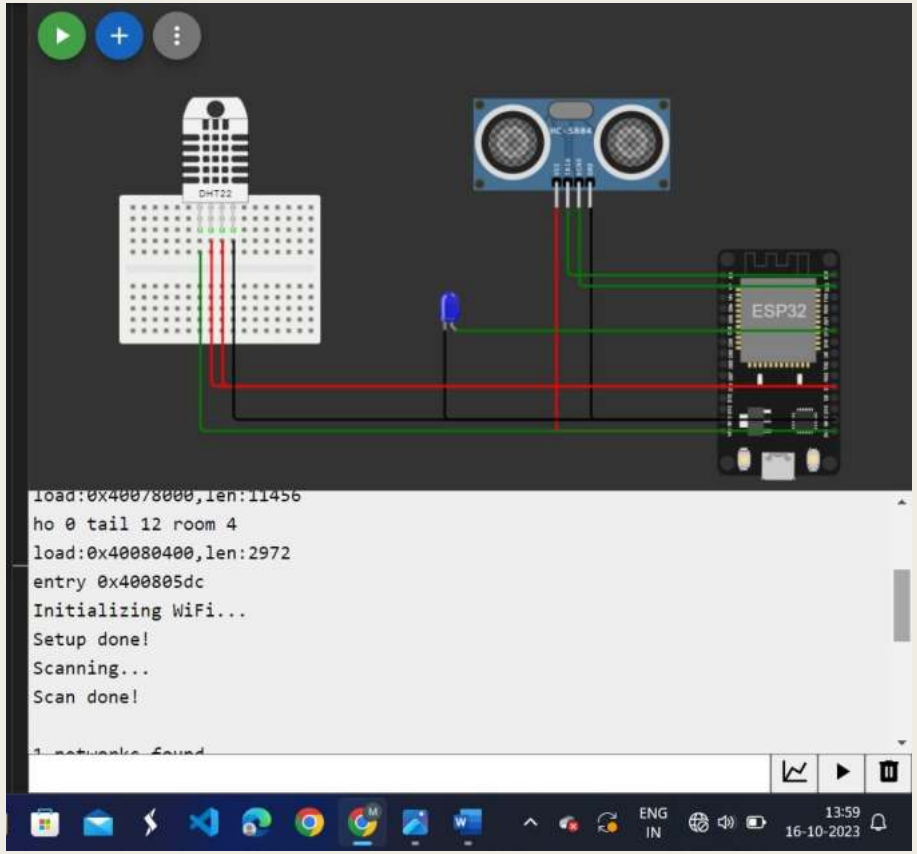
Smart water fountains are innovative devices combine water filtration with advanced technology. They offer touchless, hands-free operation and often feature sensors for user interaction and maintenance alerts. Connectivity options enable remote monitoring, data analytics, and customization. These fountains are ideal for environments like offices and public spaces, promoting water quality, hygiene, and sustainability. They also encourage the use of reusable containers, reducing the need for single-use plastic bottles.

TINKERCAD

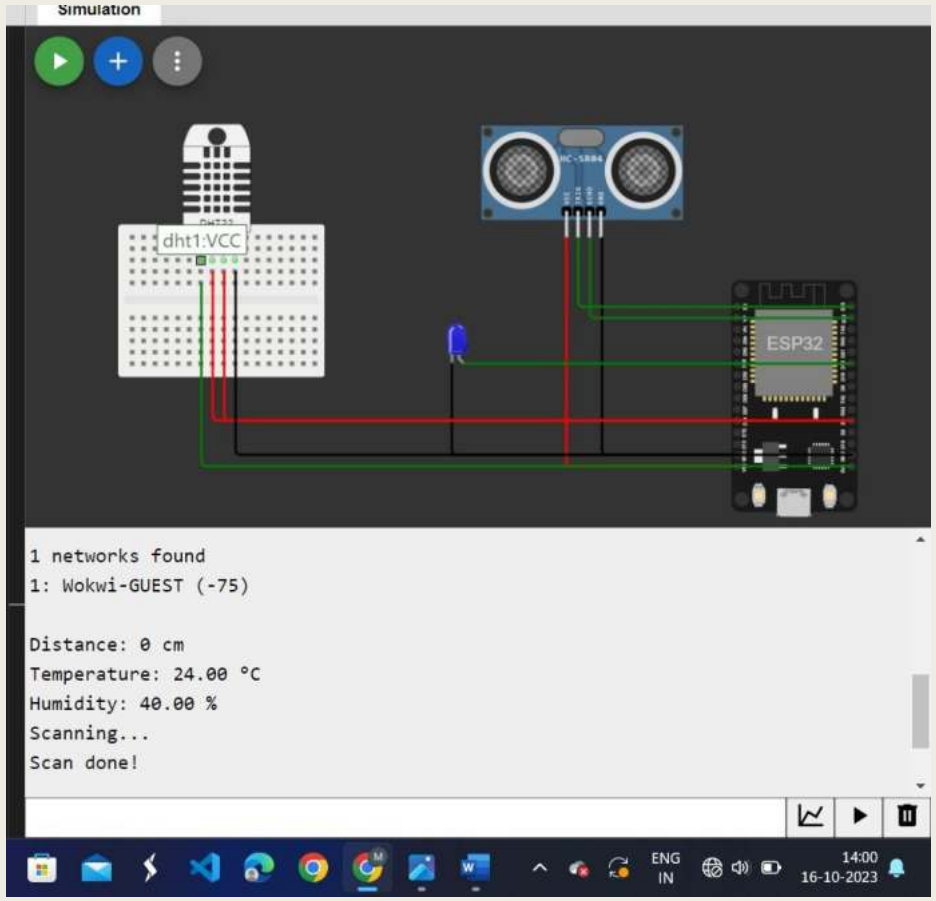


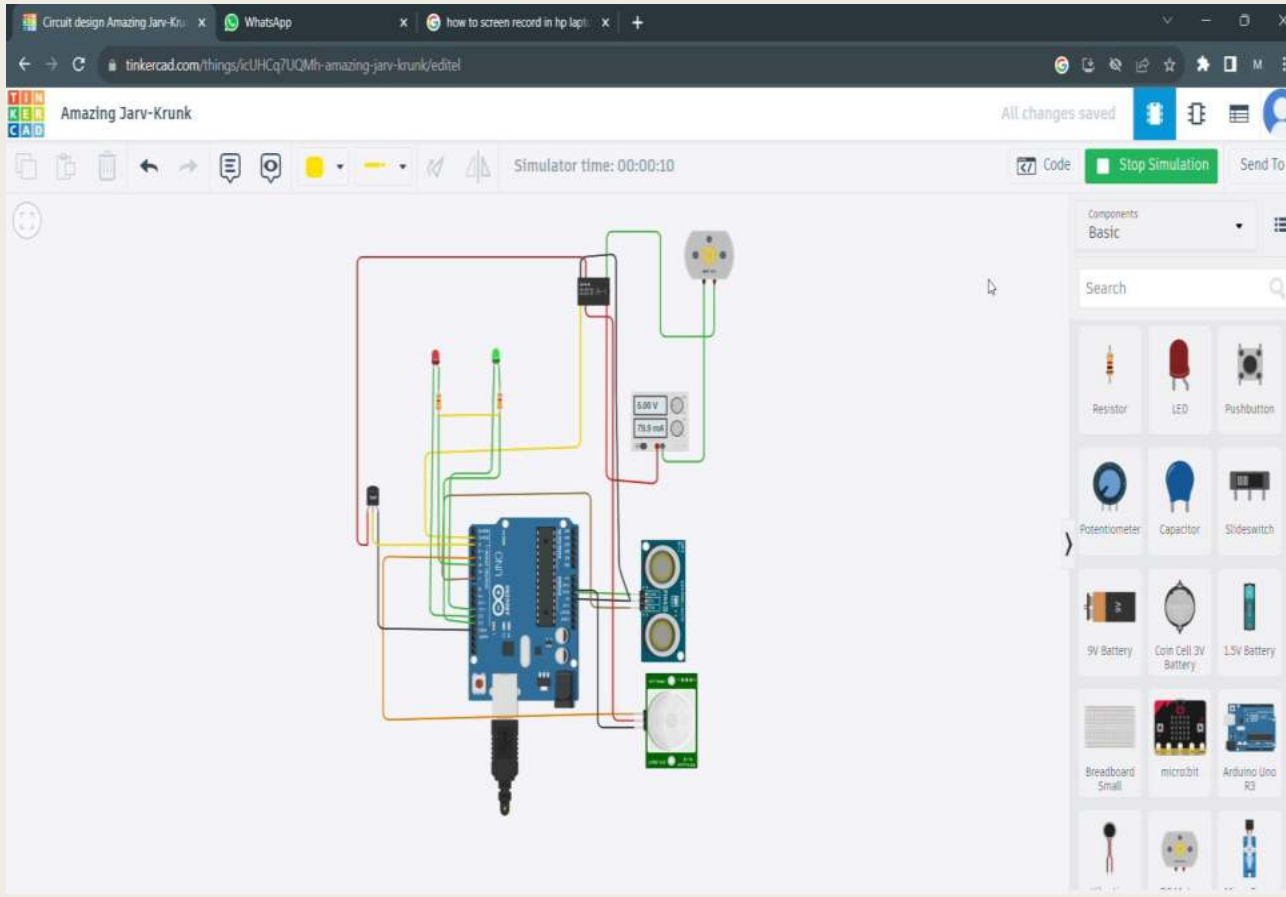
wokwi


Simulation
DHT22
HC-SR04
ESP32
ets Jul 29 2019 12:21:46
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:1156
load:0x40078000,len:11456
ho 0 tail 12 room 4
ENG
IN
13:59
16-10-2023

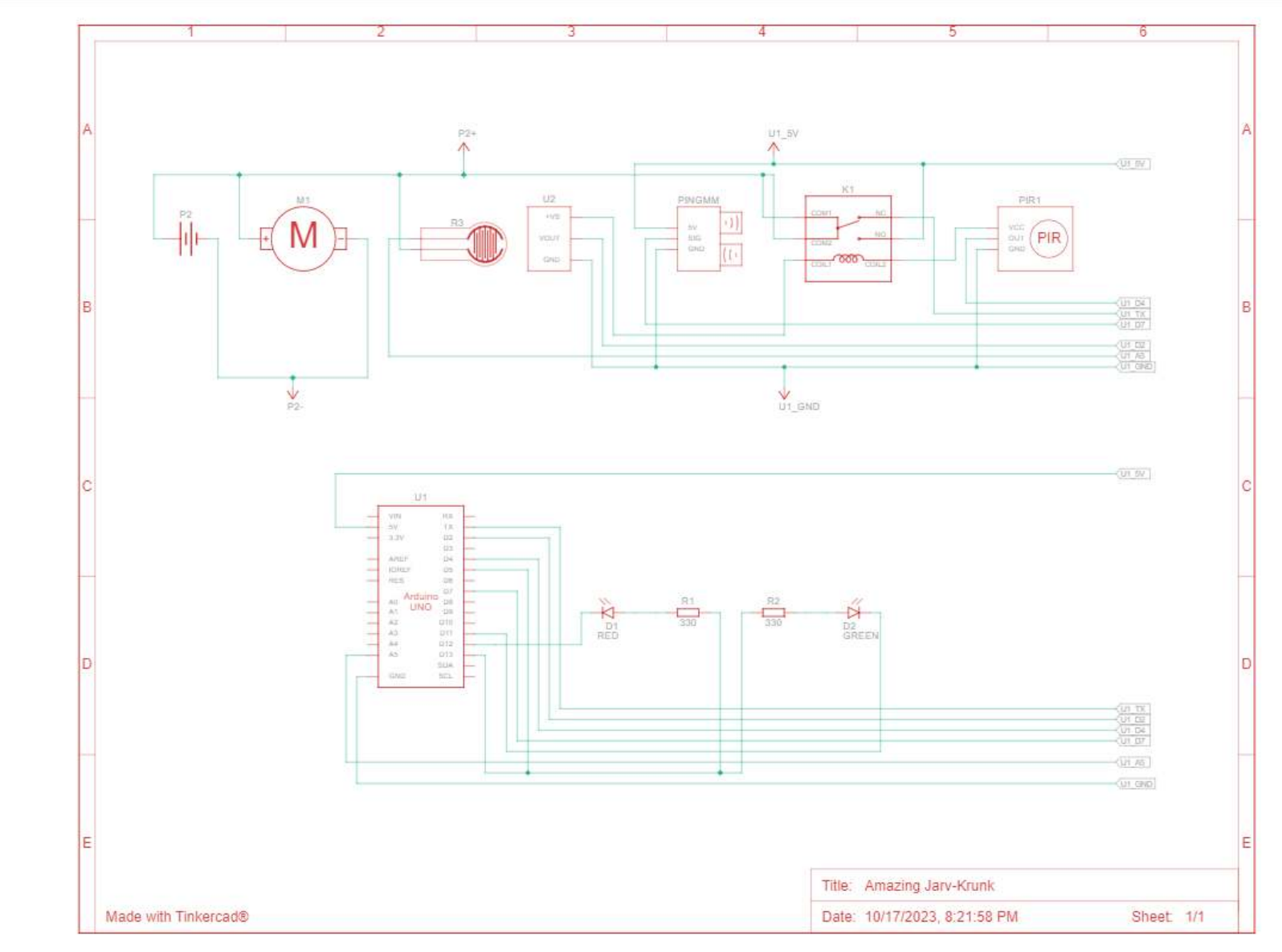

DHT22
HC-SR04
ESP32
load:0x40078000,len:11456
ho 0 tail 12 room 4
load:0x40080400,len:2972
entry 0x400805dc
Initializing WiFi...
Setup done!
Scanning...
Scan done!
ENG
IN
13:59
16-10-2023

Output in video(tinkercad)





CIRCUIT DIAGRAM

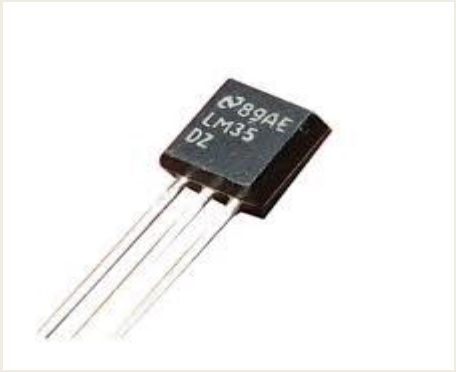


ULTRASONIC SENSOR

An ultrasonic sensor is an instrument that measures the distance to an object using ultrasonic sound waves. An ultrasonic sensor uses a transducer to send and receive ultrasonic pulses that relay back information about an object's proximity

TEMPERATURE SENSOR

A temperature sensor is a device used to measure temperature. This can be air temperature, liquid temperature or the temperature of solid matter. There are different types of temperature sensors available and they each use different technologies and principles to take the temperature measurement.



PIR SENSOR

A passive infrared sensor (PIR sensor) is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view. They are most often used in PIR-based motion detectors. PIR sensors are commonly used in security alarms and automatic lighting applications.

PROGRAM

ULTRASONIC SENSOR

```
import machine
mport 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()

 # 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

 # 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
```

TEMPERATURE SENSOR

```
import Adafruit_DHT
 # Set the sensor type, DHT22 or DHT11
 sensor = Adafruit_DHT.DHT22
 # GPIO pin where the data pin of the sensor is connected
pin = 4 # Change this to the appropriate GPIO pin on your Raspberry Pi
 try:
:# Try to grab a sensor reading.
 humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)
if humidity is not None and temperature is not None:
print(f"Temperature: {temperature:.1f}°C")
print(f'Humidity: {humidity:.1f}%")
 else:
print("Failed to retrieve data from the sensor. Check your connections.")
except Exception as e:
print(f"Error: {str(e)}")
```

EPS32 wifi module

```
import network
import urequests
# WiFi configuration
SSID = "Your_SSID"
PASSWORD = "Your_Password"
# URL to fetch data from
URL = "http://example.com" # Replace with the actual URL
 # Connect to WiFi
sta_if = network.WLAN(network.STA_IF)
sta_if.active(True)
sta_if.connect(SSID, PASSWORD)
# Wait until the ESP32 is connected to the WiFi network
 while not sta_if.isconnected():
pass
print("Connected to WiFi")
# Fetch data from the specified URL
try:
response = urequests.get(URL)
if response.status_code == 200:
print("HTTP Status Code 200: OK")
print("Content:")
print(response.text) else:
 print(f"HTTP Status Code {response.status_code}")
 except Exception as e:
print("An error occurred:", e)
finally:
 response.close()
```

THANK YOU