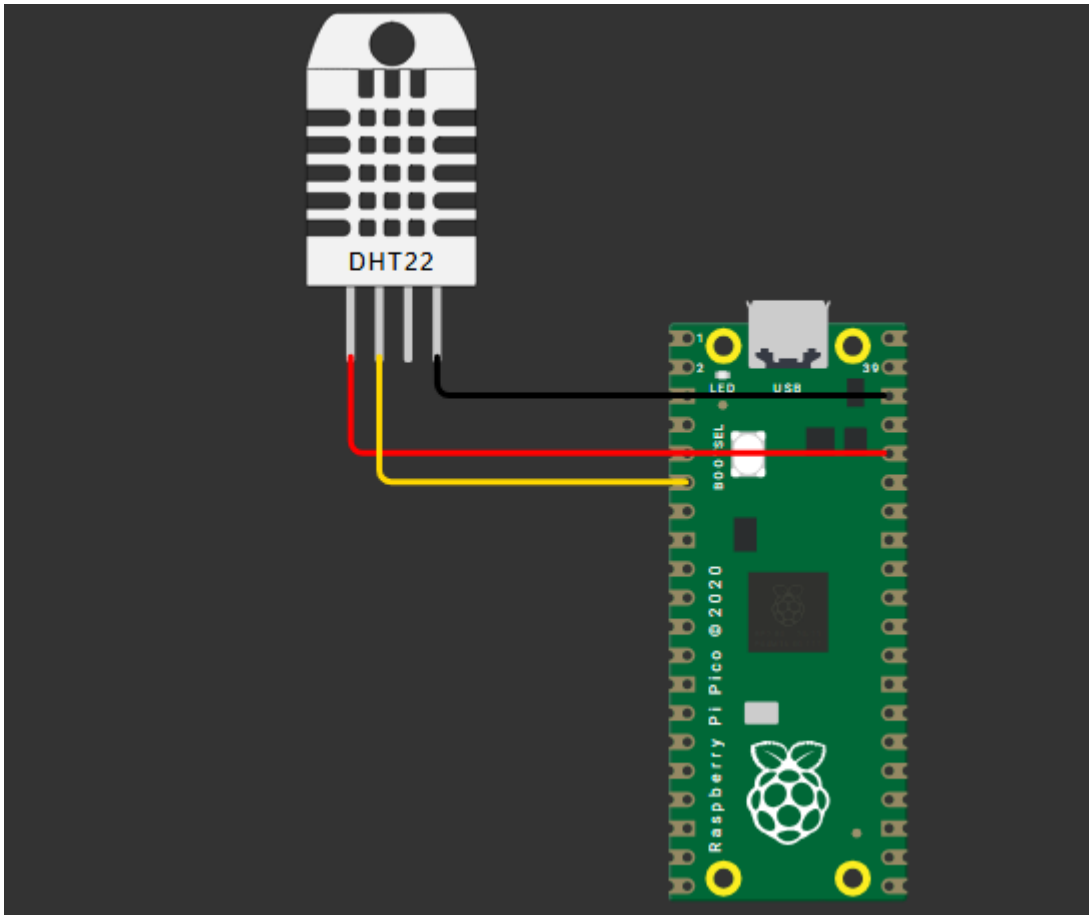
PRACTICAL 2:

Understanding the connectivity of Raspberry-Pi /Beagle board circuit with temperature sensor. Write an application to read the environment temperature. If temperature crosses a threshold value, generate alerts using LEDs.

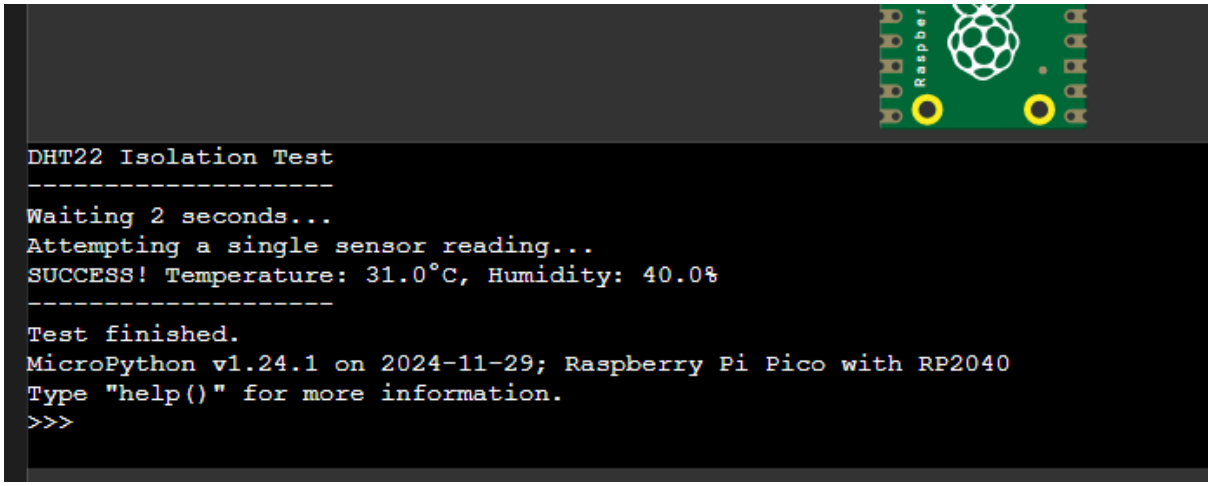


PROGRAM:

```
# Paste this code into main.py and run it

import time

from machine import Pin

import dht


dht_pin = Pin(4, Pin.IN, Pin.PULL_UP)

dht_device = dht.DHT22(dht_pin)
```

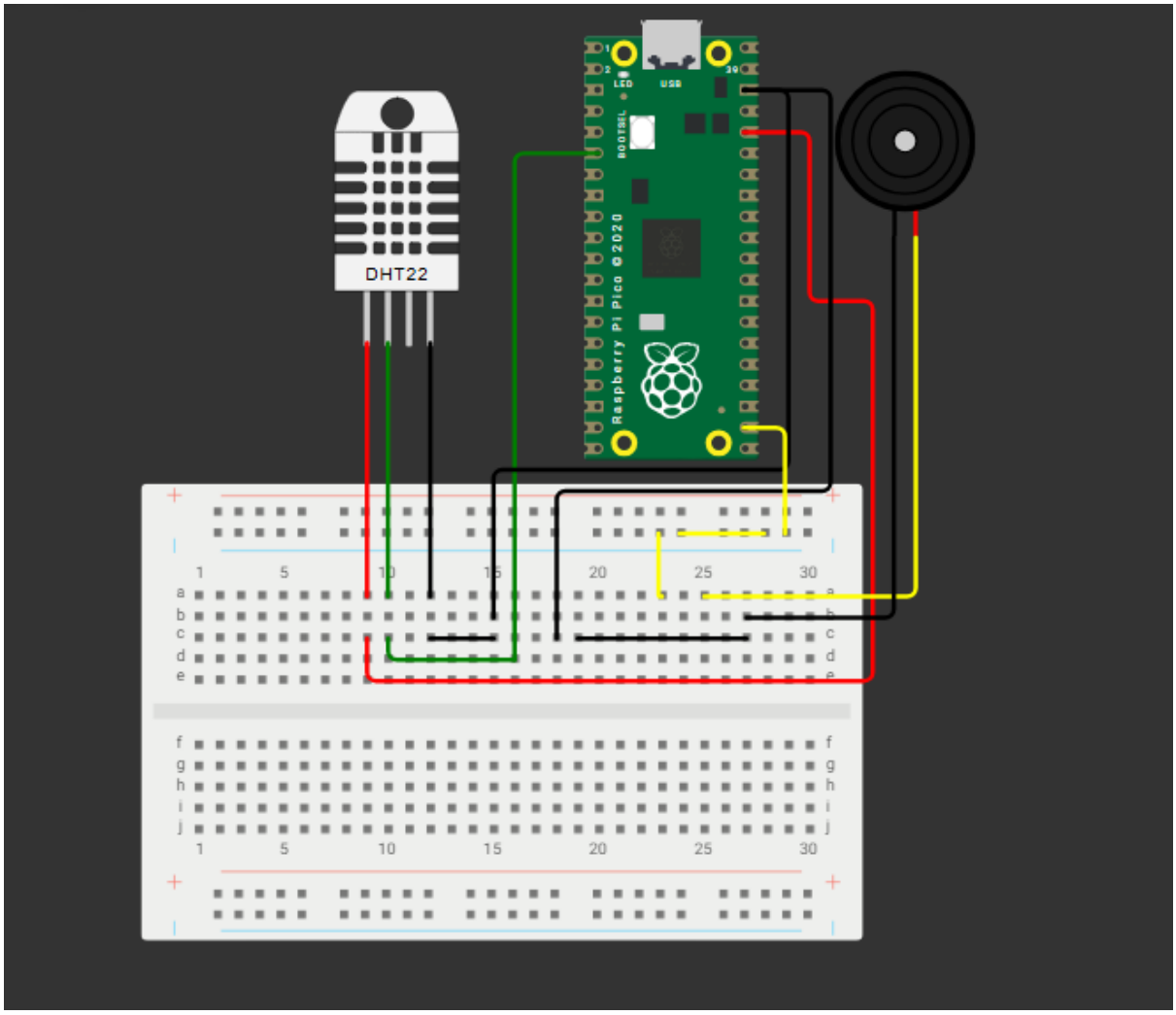
```
print("DHT22 Isolation Test")
print("--------------------")
print("Waiting 2 seconds...")

time.sleep(2)
print("Attempting a single sensor reading...")

try:
    dht_device.measure()
    time.sleep(0.5)
    temp = dht_device.temperature()
    hum = dht_device.humidity()
    print(f"SUCCESS! Temperature: {temp}°C, Humidity: {hum}%")
except Exception as e:
    print(f"FAILURE! The sensor is not responding. Error: {e}")

print("--------------------")
print("Test finished.")
```

OUTPUT:



ASSIGNMENT 2

Create a circuit using Raspberry Pi, DHT11 and Buzzer. Write a program when the temperature goes beyond 35 degrees, the buzzer will start ringing.



PROGRAM:

```
import time
from machine import Pin
import dht


# --- CONFIGURATION ---
# The GPIO pin the DHT11 data line is connected to.
```

```
dht_pin = Pin(4, Pin.IN, Pin.PULL_UP)

# The GPIO pin the buzzer is connected to.
buzzer_pin = Pin(17, Pin.OUT)

# The temperature threshold in Celsius.
THRESHOLD_TEMP = 35.0

# --- HARDWARE INITIALIZATION ---
# Initialize the DHT11 sensor.
dht_device = dht.DHT11(dht_pin)

# Give the sensor a moment to settle.
print("Waiting for sensor to initialize...")
time.sleep(1)

# --- MAIN APPLICATION LOGIC ---
print("Temperature Alert System Started for Pico. Press CTRL+C to exit.")

while True:
    try:
        # In MicroPython, you must call measure() first.
        dht_device.measure()

        # Then you can read the temperature.
        temperature_c = dht_device.temperature()

        print(f"Current Temperature: {temperature_c:.1f}°C")

        # --- THRESHOLD CHECK AND ALERT ---
```

```
        if temperature_c > THRESHOLD_TEMP:
            # Temperature is above the threshold, turn the buzzer ON.
            buzzer_pin.value(1)  # 1 is ON
            print(f"ALERT! Temperature ({temperature_c:.1f}°C) is above
{THRESHOLD_TEMP}°C. BUZZER ON.")
        else:
            # Temperature is normal, turn the buzzer OFF.
            buzzer_pin.value(0)  # 0 is OFF
            print("Temperature is normal. Buzzer is OFF.")

    except Exception as error:
        # Errors happen fairly often with DHT sensors.
        print(f"Failed to read sensor: {error}")

    # Wait 2 seconds before the next reading.
    time.sleep(2.0)
```

OUTPUT: