**WOKWI online simulator Output**

ESP32 proceesor is integrated with Ultrasonic sensor HC-SR04 ultrasonic sesnor to find the object distance from the sensor. Python code to find distance of object and its simulated output

from machine import Pin, time\_pulse\_us

import time

# Define the pins for the Ultrasonic Sensor

TRIG\_PIN = 23 # GPIO pin for TRIG

ECHO\_PIN = 22 # GPIO pin for ECHO

# Set up the pins

trigger = Pin(TRIG\_PIN, Pin.OUT)

echo = Pin(ECHO\_PIN, Pin.IN)

# Function to measure distance

def measure\_distance():

# Send a 10us pulse to trigger the sensor

trigger.value(0)

time.sleep\_us(2)

trigger.value(1)

time.sleep\_us(10)

trigger.value(0)

# Measure the duration of the echo pulse

duration = time\_pulse\_us(echo, 1)

# Calculate the distance in centimeters

distance = duration \* 0.0343 / 2 # Speed of sound is 343m/s, divided by 2 because the pulse travels out and back

return distance

# Main loop

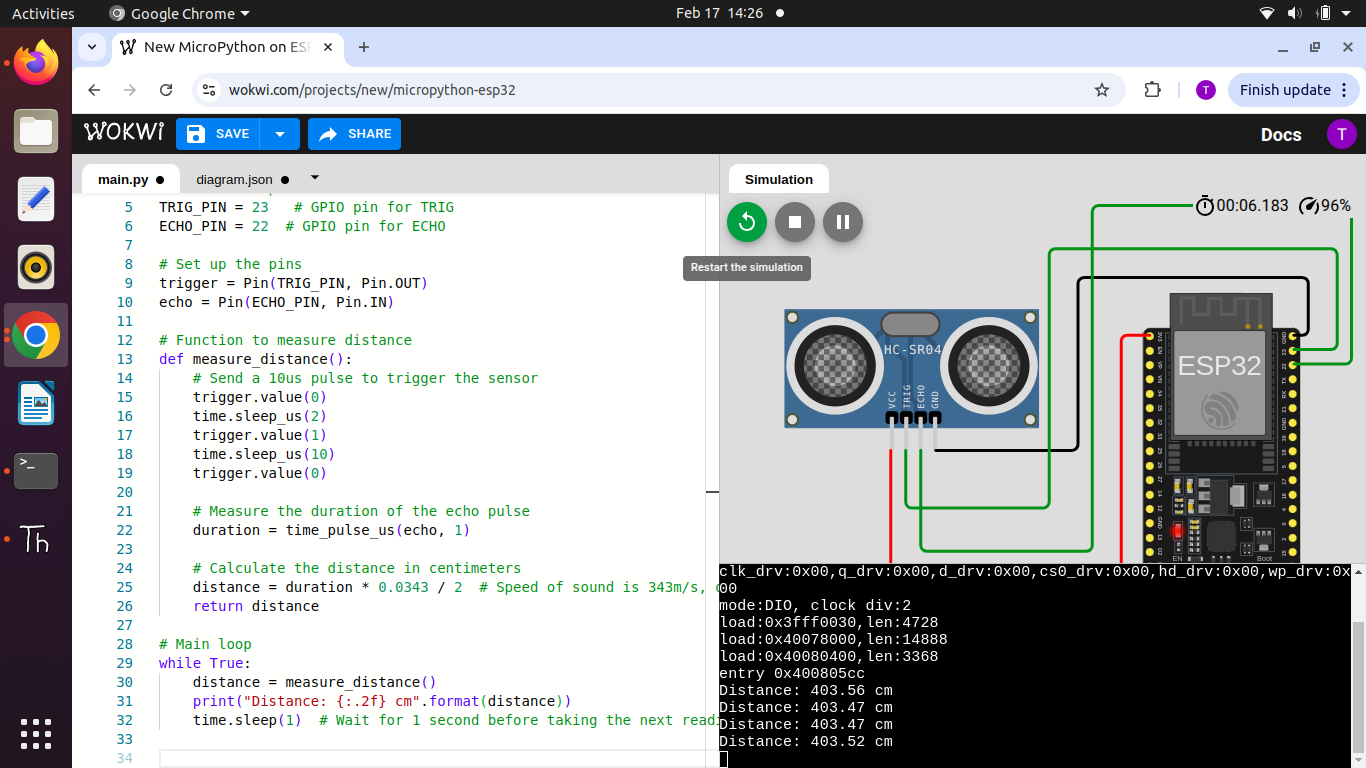
while True:

distance = measure\_distance()

print("Distance: {:.2f} cm".format(distance))

time.sleep(1) # Wait for 1 second before taking the next reading

**Output:**



Arduino IDE code to find the distance of the object using ESP32 processor and ultrasonic sensor and its output

#define TRIG\_PIN 23 // GPIO pin for Trigger

#define ECHO\_PIN 22 // GPIO pin for Echo

void setup()

{

Serial.begin(115200); // Start Serial Monitor

pinMode(TRIG\_PIN, OUTPUT);

pinMode(ECHO\_PIN, INPUT);

}

void loop()

{

long duration;

float distance;

// Send a 10-microsecond pulse to trigger the ultrasonic sensor

digitalWrite(TRIG\_PIN, LOW);

delayMicroseconds(2);

digitalWrite(TRIG\_PIN, HIGH);

delayMicroseconds(10);

digitalWrite(TRIG\_PIN, LOW);

// Measure the echo response time

duration = pulseIn(ECHO\_PIN, HIGH);

// Convert the duration to distance (speed of sound = 343m/s)

distance = (duration \* 0.0343) / 2;

// Print distance to Serial Monitor

Serial.print("Distance: ");

Serial.print(distance);

Serial.println(" cm");

delay(1000); // Wait 1 second before next reading

}

**Output:**

