**ASSIGNMENT 3**

Name: Sanved Bahendwar

SUID: 484052966

WOKWI

import network

import time

import urandom

from umqtt.simple import MQTTClient

# ThingSpeak MQTT broker details

mqtt\_client\_id = "GQECEjshJic9LywZPTkmAzI"

mqtt\_user = "GQECEjshJic9LywZPTkmAzI"

mqtt\_password = "rbS4eaZWlr53zjeHvhKKhrNs"

mqtt\_server = "mqtt3.thingspeak.com"

mqtt\_port = 1883

mqtt\_topic\_temperature = "channels/2488588/publish/fields/field1"

mqtt\_topic\_humidity = "channels/2488588/publish/fields/field2"

mqtt\_topic\_co2 = "channels/2488588/publish/fields/field3"

# Wi-Fi details

WIFI\_SSID = "Wokwi-GUEST"

WIFI\_PASSWORD = ""

# Historical data storage

historical\_data = []

# Function to generate random sensor values

def generate\_sensor\_data():

temperature = urandom.uniform(-50, 50)

humidity = urandom.uniform(0, 100)

# Ensure CO2 value is within the acceptable range (300 to 2000 ppm)

co2 = urandom.uniform(300, 2000)

return temperature, humidity, co2

# Function to publish data to ThingSpeak

def publish\_to\_thingspeak(temperature, humidity, co2):

client = MQTTClient(mqtt\_client\_id, mqtt\_server, user=mqtt\_user, password=mqtt\_password)

client.connect()

client.publish(mqtt\_topic\_temperature, str(temperature))

client.publish(mqtt\_topic\_humidity, str(humidity))

client.publish(mqtt\_topic\_co2, str(co2))

client.disconnect()

# Connect to Wi-Fi

sta\_if = network.WLAN(network.STA\_IF)

sta\_if.active(True)

sta\_if.connect(WIFI\_SSID, WIFI\_PASSWORD)

# Wait for Wi-Fi connection

while not sta\_if.isconnected():

pass

print("Connected to Wi-Fi")

# Main loop to generate and publish sensor data

while True:

temperature, humidity, co2 = generate\_sensor\_data()

historical\_data.append((temperature, humidity, co2)) # Store historical data

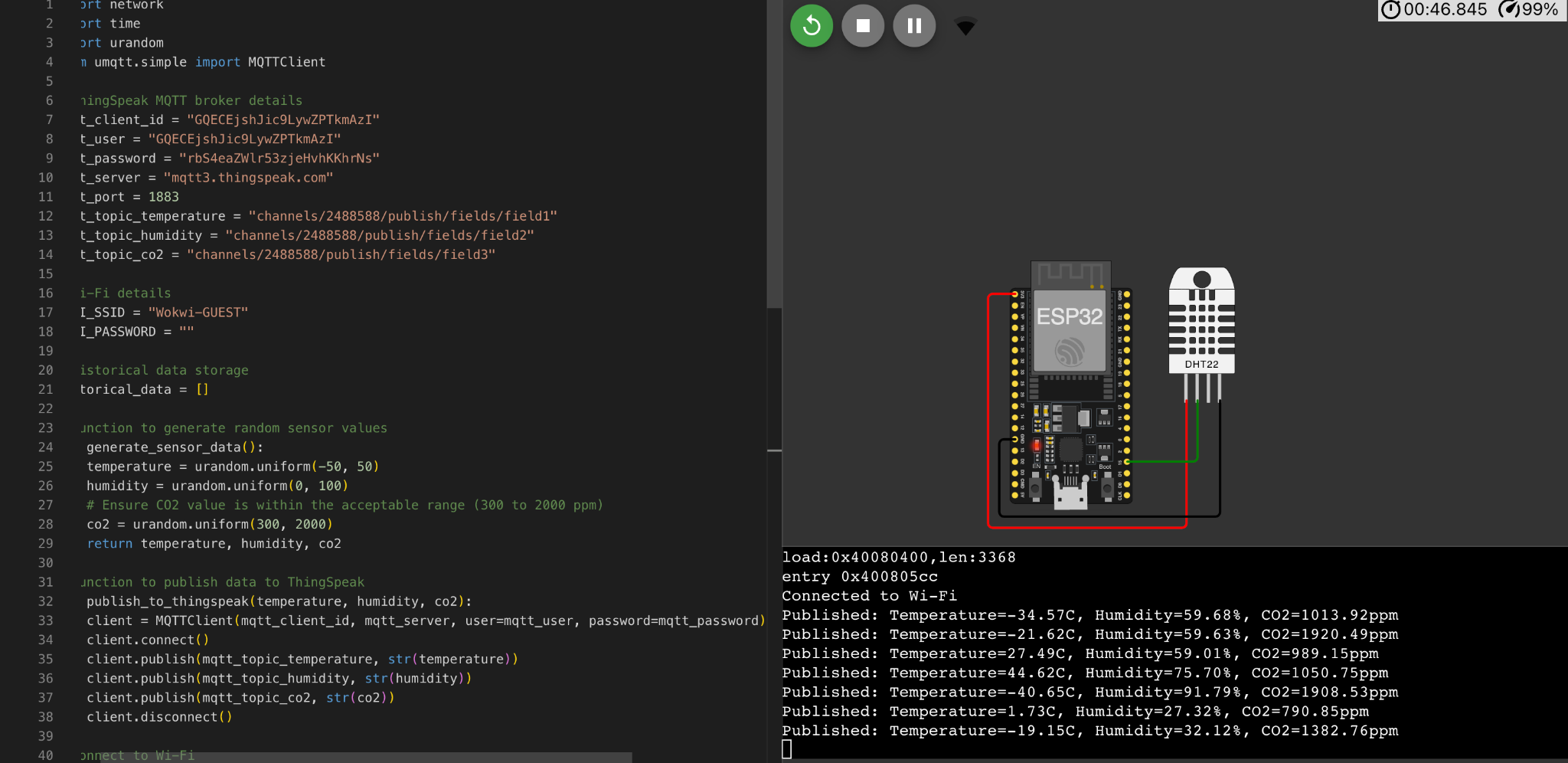
if len(historical\_data) > 720: # Approximately 5 hours with data every 5 seconds

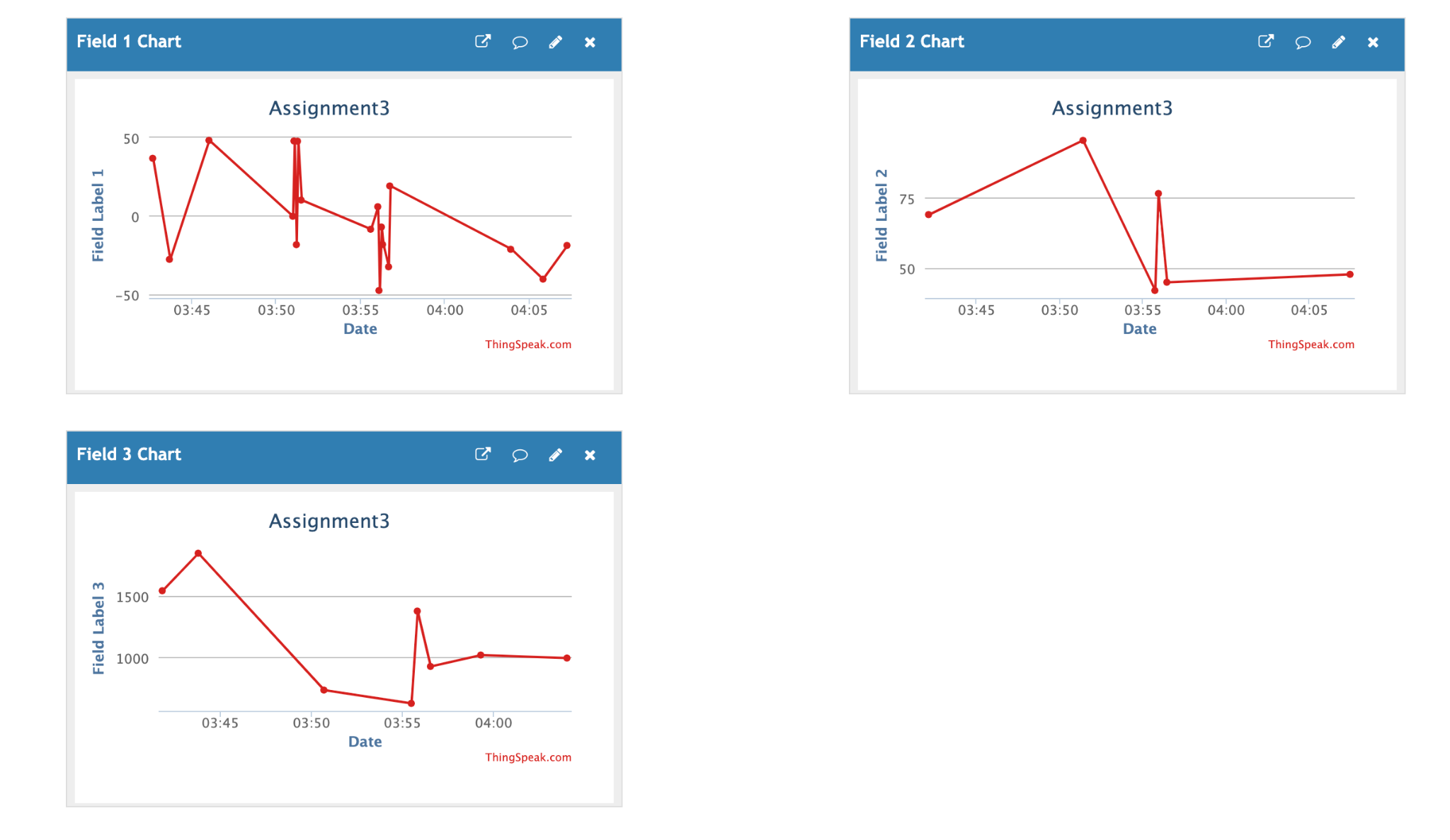
historical\_data.pop(0) # Remove oldest data point if exceeds 5 hours

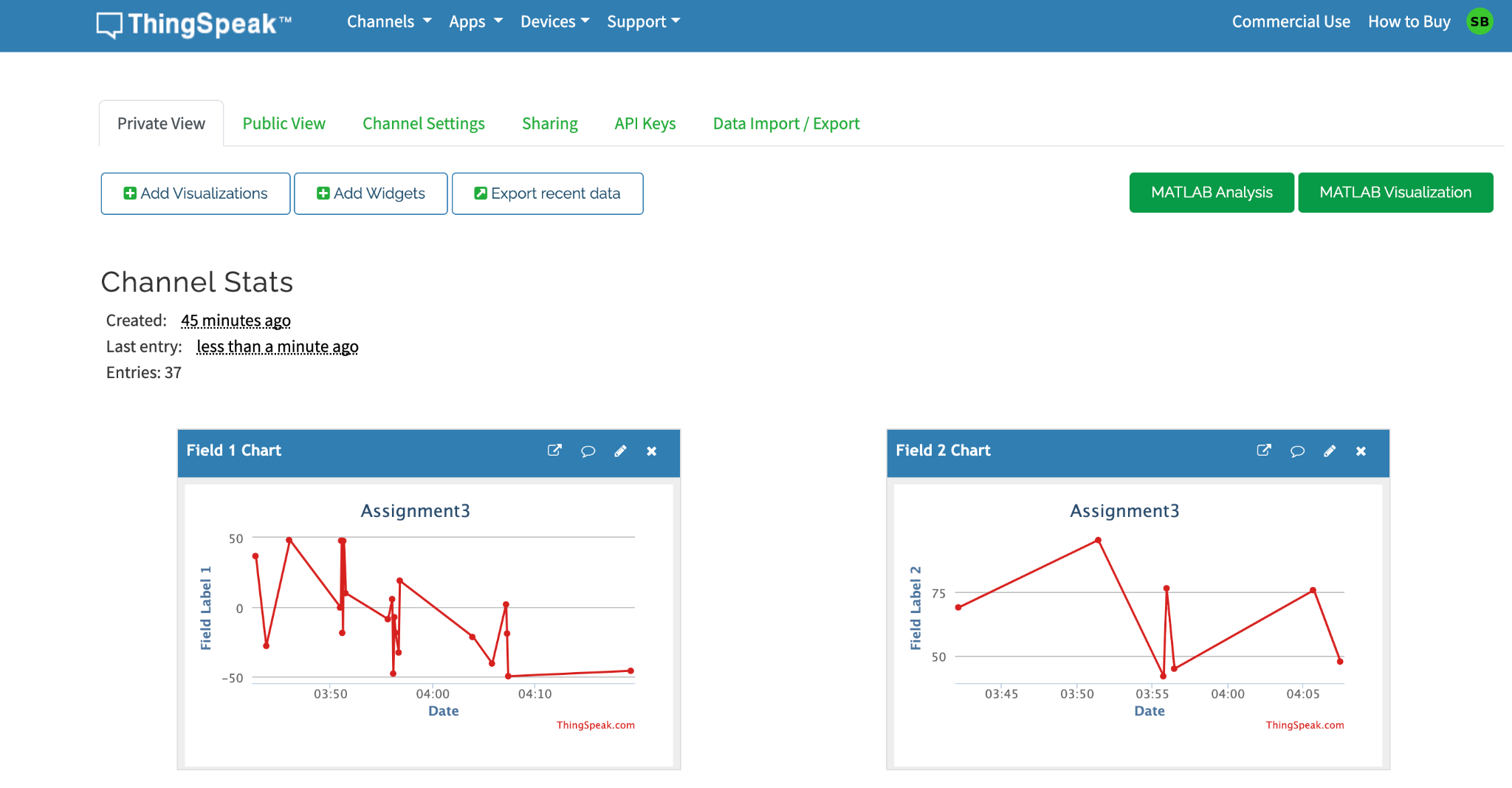
publish\_to\_thingspeak(temperature, humidity, co2)

print("Published: Temperature={:.2f}C, Humidity={:.2f}%, CO2={:.2f}ppm".format(temperature, humidity, co2))

time.sleep(5) # Adjust the delay as needed (Reduced to 5 seconds for faster data entry)







MATLAB ANALYSIS

% Set your ThingSpeak channel ID and read API key

channelID = 2488588;

readAPIKey = 'XDZ001M5HV402EE5';

% Get the current time and time five hours ago

currentTime = datetime('now', 'TimeZone', 'UTC');

fiveHoursAgo = currentTime - hours(5);

% Set up the ThingSpeak URL for fetching data

url = sprintf('https://api.thingspeak.com/channels/%d/feeds.json?api\_key=%s&start=%s&end=%s', ...

channelID, readAPIKey, datestr(fiveHoursAgo, 'yyyy-mm-ddTHH:MM:SSZ'), ...

datestr(currentTime, 'yyyy-mm-ddTHH:MM:SSZ'));

% Fetch data from ThingSpeak

data = webread(url);

% Extract sensor data

if ~isempty(data.feeds)

sensorData = [data.feeds.field1]; % Assuming the sensor data is in Field 1

timestamps = datetime({data.feeds.created\_at}, 'InputFormat', 'yyyy-MM-dd''T''HH:mm:ss''Z''', 'TimeZone', 'UTC');

% Display sensor data

disp('Sensor Data:');

disp(sensorData);

disp('Timestamps:');

disp(timestamps);

else

disp('No data found in the specified time range.');

end

Sensor Data:

36.35374-28.0464547.79149-0.594830547.38763-18.6835847.240279.70949-8.8564995.511701-47.8815-7.433379-18.57623-32.7976218.73416-21.61746-40.650281.729667-19.15468-49.731-18.58542-45.9010115.2933546.2425-23.22364

Timestamps:

Columns 1 through 8

27-Mar-2024 22:11:50 27-Mar-2024 22:12:09 27-Mar-2024 22:12:41 27-Mar-2024 22:13:42 27-Mar-2024 22:13:48 27-Mar-2024 22:16:01 27-Mar-2024 22:20:42 27-Mar-2024 22:20:59

Columns 9 through 16

27-Mar-2024 22:21:05 27-Mar-2024 22:21:12 27-Mar-2024 22:21:18 27-Mar-2024 22:21:24 27-Mar-2024 22:21:30 27-Mar-2024 22:25:30 27-Mar-2024 22:25:37 27-Mar-2024 22:25:44

Columns 17 through 24

27-Mar-2024 22:25:50 27-Mar-2024 22:25:56 27-Mar-2024 22:26:02 27-Mar-2024 22:26:08 27-Mar-2024 22:26:15 27-Mar-2024 22:26:21 27-Mar-2024 22:26:27 27-Mar-2024 22:26:33

Columns 25 through 32

27-Mar-2024 22:26:40 27-Mar-2024 22:26:46 27-Mar-2024 22:29:19 27-Mar-2024 22:33:57 27-Mar-2024 22:34:04 27-Mar-2024 22:35:42 27-Mar-2024 22:35:49 27-Mar-2024 22:37:10

Columns 33 through 40

27-Mar-2024 22:37:16 27-Mar-2024 22:37:22 27-Mar-2024 22:37:28 27-Mar-2024 22:49:11 27-Mar-2024 22:49:25 27-Mar-2024 22:50:15 27-Mar-2024 22:50:21 27-Mar-2024 22:50:57

Columns 41 through 43

27-Mar-2024 22:53:34 27-Mar-2024 22:53:45 27-Mar-2024 23:06:56

