9238-Mangayarkarasi College Of Engineering

(Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai)

MANGAYARKARASI NAGAR, PARAVAI, MADURAI – 625 402

Website: <http://mce-madurai.ac.in>

E-Mail: : [mangai.enggcoll@gmail.com](mailto:mangai.enggcoll@gmail.com)

AIR QUALITY MONITORING PHASE 4

Project team Members;

T. Abinaya -923821106003

R. Mahalakshmi -923821106024

D. Mangalajothi -923821106026

J. Nourin Riffana -923821106031

M. Varshini -923821106054

Project Guide

Mr.R.M. SenthilKumar AP/ECE

INTRODUCTION



Air quality monitoring using IoT (Internet of Things) is a cutting-edge technology that revolutionizes our ability to track and manage air pollution. By integrating sensor networks and data analytics, IoT air quality monitoring systems provide real-time information on various air pollutants, enabling us to make informed decisions for healthier environments and sustainable living. This technology has the potential to transform how we address air quality concerns, ensuring a cleaner and safer future for all.

Air quality is a fundamental aspect of our daily lives, impacting our health, well-being, and the environment. Poor air quality is associated with a range of health problems, including respiratory issues, cardiovascular diseases, and even premature mortality. It also contributes to environmental degradation and climate change. As the world grapples with the consequences of air pollution, technology, particularly the Internet of Things (IoT), has emerged as a powerful tool to monitor, manage, and improve air quality.

COMPONENTS OF IOT AIR QUALITY MONITORING:

Sensors: At the heart of IOT air quality monitoring are specialized sensors designed to detect various air pollutants, such as particulate matter (PM), volatile organic compounds (VOCs), nitrogen dioxide (NO2), sulphur dioxide (SO2), and carbon monoxide (CO). These sensors can be deployed in various locations, from urban areas to industrial sites, providing real-time data on air quality.

Connectivity: IOT relies on an extensive network of wireless connectivity, such as Wi-Fi, cellular, or Low-Power Wide Area Networks (LPWAN), to transmit data from sensors to a central server or cloud-based platform. This enables remote monitoring and data accessibility from virtually anywhere.

Data Processing: The collected data is processed and analyzed using advanced algorithms and machine learning techniques. This analysis can provide insights into pollution patterns, trends, and potential health risks. Real-time data is made available to decision-makers, researchers, and the public through user-friendly interfaces.

BENEFITS OF IOT AIR QUALITY MONITORING:

Real-Time Monitoring: IOT allows for continuous, real-time monitoring of air quality, enabling timely responses to changing conditions and helping authorities implement mitigation measures swiftly.

Data Accuracy: IOT sensors provide highly accurate data, reducing the margin of error associated with traditional monitoring methods. This accuracy is crucial for making informed decisions and understanding the true impact of air pollution.

Cost-Efficiency: IOT technologies can be more cost-effective than traditional air quality monitoring stations, making it feasible to deploy a wider network of sensors, especially in areas where comprehensive monitoring was previously challenging.

Customization: IOT air quality monitoring systems can be tailored to specific needs, such as urban areas, industrial zones, or indoor spaces. This adaptability ensures that the data collected is relevant to the context.

SIMULATION PROCESS

WOKWI:

Wokwi is an embedded systems and IOT simulator supporting ESP32, Arduino, and the Raspberry Pi Pico. Your code never leaves your computer – Wokwi runs the simulation inside VS Code, using the firmware binaries from your project.

Wokwi is a platform that allows you to simulate And test your code for microcontroller-based projects, including Those written in Python for microcontrollers like the ESP32. To Implement an air quality monitoring system in a environment using Wokwi, we’ll need to follow these steps:

1.Create a Wokwi Account:

Start by creating an account on the Wokwi platform if you Don’t have one already.

2.Select the Microcontroller:

In Wokwi, choose the microcontroller you want to work With. For air quality monitoring projects, the ESP32 is a Popular choice due to its built-in Wi-Fi capabilities.

3.Design Your Circuit:

Using Wokwi’s intuitive drag-and-drop interface, design the Circuit for your air quality monitoring system. This may Include adding sensors (e.g., DHT11 for temperature and Humidity), IR sensor and any other components you need.

4.Write the Python Code:

In the Wokwi interface, you can write Python code to Interact with the sensors and control the microcontroller. For example, you can use the machine module to Configure pins and sensors and the request’s module to Send data to a server or ThingSpeak.

5.Simulate Your Project:

Click the “Run” button to simulate your project. You can Observe how your Python code interacts with the virtual sensors, and microcontroller.

6.Test and Debug:

Use the simulation to test your code for Air quality monitoring. You can check if the temperature And humidity readings are correct and if your data sending Function works as expected. If any issues arise, use the Debug tools provided by Wokwi to identify and resolve Problems in your code.

7.Save Your Project:

Save your project on Wokwi so you can access it later or Share it with others.

8.Explore More Sensors and Components:

You can expand your project by adding more sensors and Components to simulate a comprehensive air quality monitoring system. For example, you can add sensors for light, or GPS to gather more data.

PYTHON CODE FOR WOKWI

The provided python program is designed for an ESP8266 based to Monitor temperature and humidity using a DHT11 sensor and send that Data to ThingSpeak, a cloud – based IOT platform .

Import machine

Import time

Import network

Import dht

Import urequests

# DHT22 data pin connected to GPIO5 (D1)

Dht\_pin = machine.Pin(23)

# IR sensor pin

Ir\_pin = machine.Pin(15)

# WiFi credentials

Ssid = “Wokwi-GUEST”

Password = “”

# ThingSpeak API key

Api\_key = “R4PC4HW83LV9O3KE”

# ThingSpeak server

Ts\_server = “api.thingspeak.com”

Channel\_number = 2319907

# Connect to WiFi

Def connect\_wifi():

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

If not sta\_if.isconnected():

Print(“Connecting to WiFi…”)

Sta\_if.active(True)

Sta\_if.connect(ssid, password)

While not sta\_if.isconnected():

Pass

Print(“Connected to WiFi”)

# Function to read and return temperature, humidity, and IR data

Def read\_sensors():

Dht22 = dht.DHT22(dht\_pin) # Create the DHT22 object

Dht22.measure()

Temp, hum = dht22.temperature(), dht22.humidity()

Ir\_data = ir\_pin.value() # Read IR data

Return temp, hum, ir\_data

# Send data to ThingSpeak

Def send\_to\_thingspeak(temp, hum, ir\_data):

Base\_url = http://{}/update?api\_key={}&field1={}&field2={}&field3={}.format(

Ts\_server, api\_key, temp, hum, ir\_data)

Response = urequests.get(base\_url)

If response.status\_code == 200:

Print(“Data sent to ThingSpeak successfully”)

Else:

Print(“Failed to send data to ThingSpeak”)

# Main program

Def main():

Connect\_wifi()

While True:

Temp, hum, ir\_data = read\_sensors()

Print(“Temperature: {}°C, Humidity: {}%, IR Data: {}”.format(temp, hum, ir\_data))

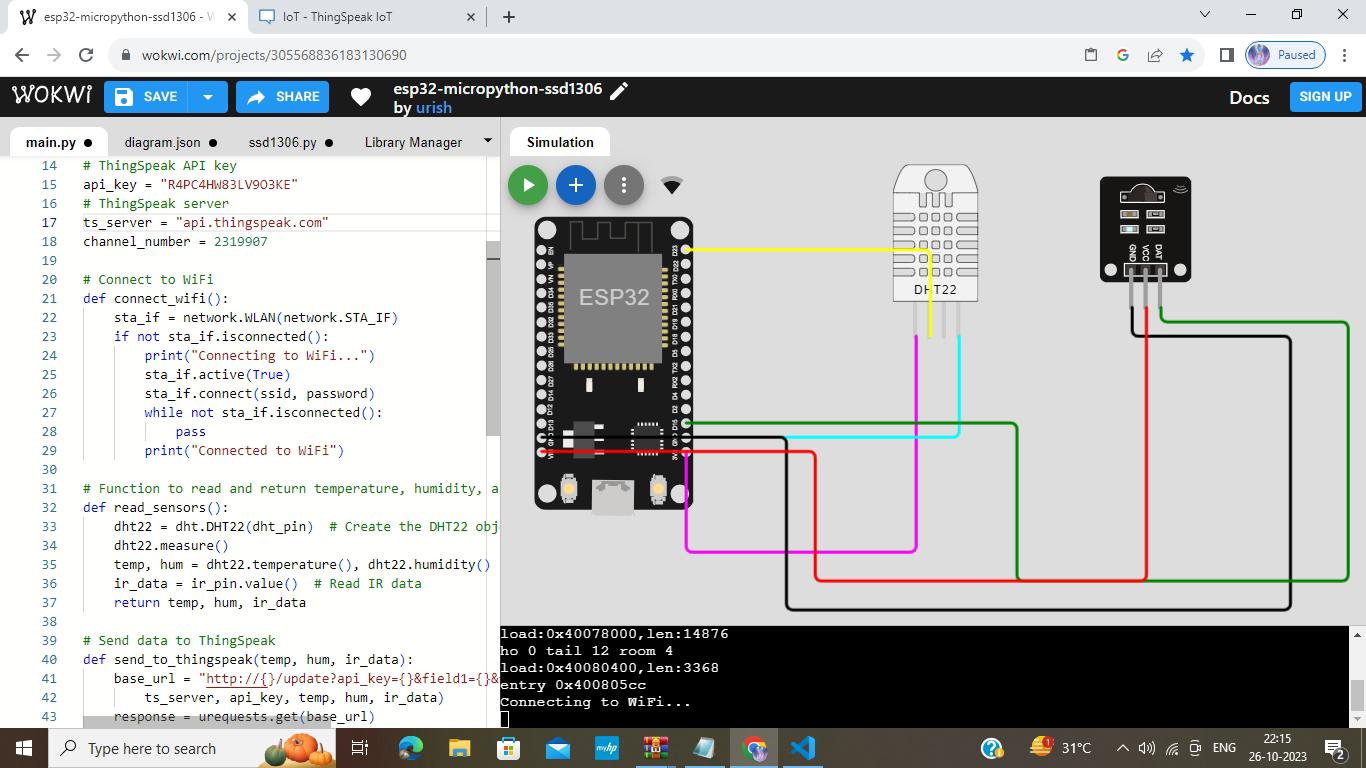
Send\_to\_thingspeak(temp, hum, ir\_data)

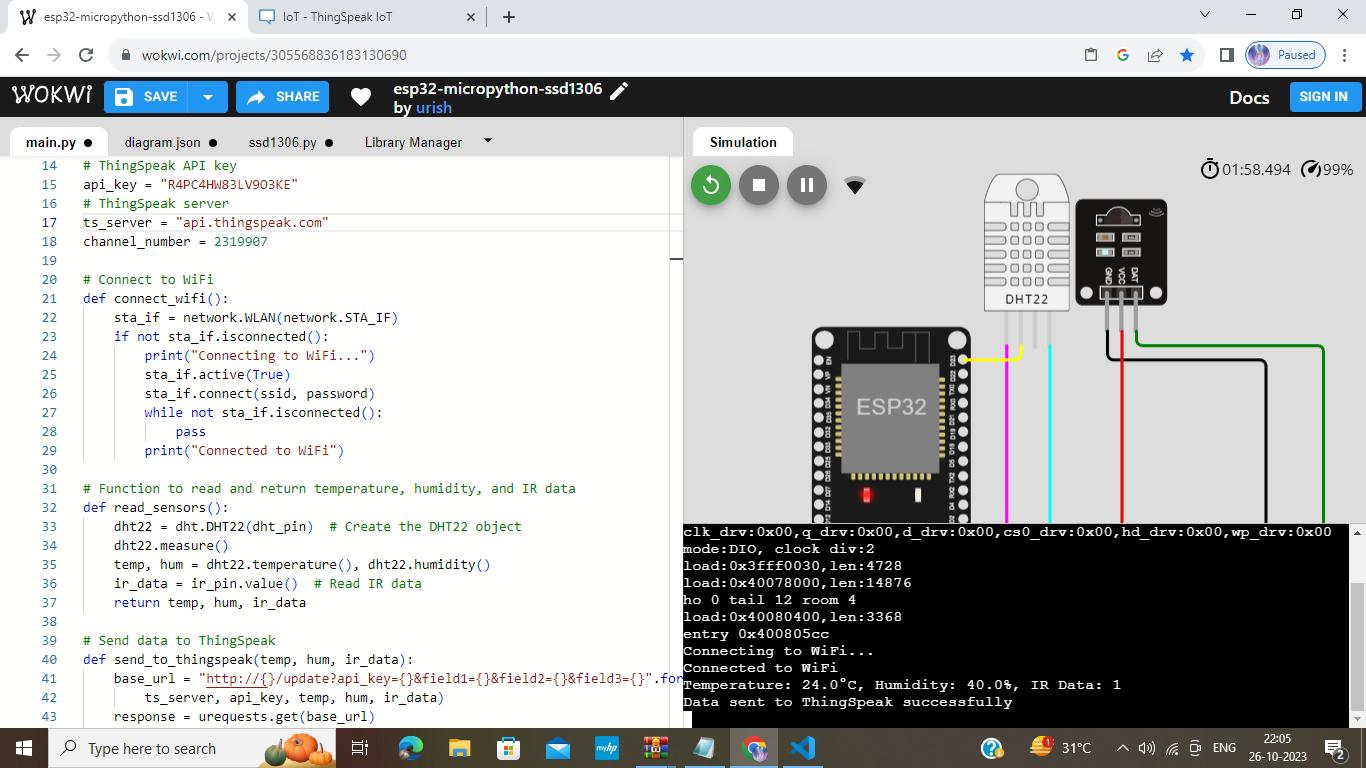
Time.sleep(300) # Send data every 5 minutes

If \_\_name\_\_ == ‘\_\_main\_\_’:

Main()

The Above Program Send the data to the Thingspeak server. The Data Get From the Microcontroller ESP32.Humdity and Temperature Can be Get using the DHT 22 Sensor. The Data will Send for Every 5 Minutes The Program Access The ThingSpeak server Via the API key.

Output from wokwi



We can adjust the Temperature and Humidity level Of the sensor because this is the simulation process Not having a physical components The Following Library Files are used in WOKWI are,

* DHT22 sensor
* IR sensor
* DHT sensor library for ESPx
* WiFi
* ThingSpeak

THINGSPEAK

ThingSpeak is our chosen server for your air quality Monitoring project.

1.Set Up ThingSpeak Account:

If you haven’t already, create an account on ThingSpeak (<https://thingspeak.com/>).

2.Create a ThingSpeak Channel:

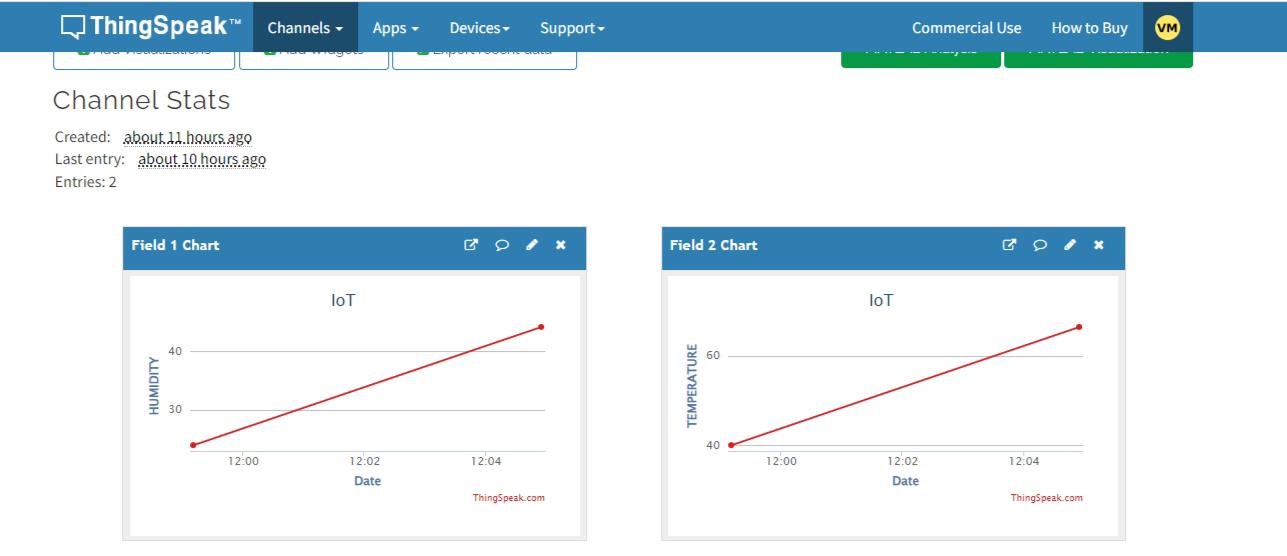
In your ThingSpeak account, create a new Channel. This channel will be used to store the data From your air quality monitoring system.

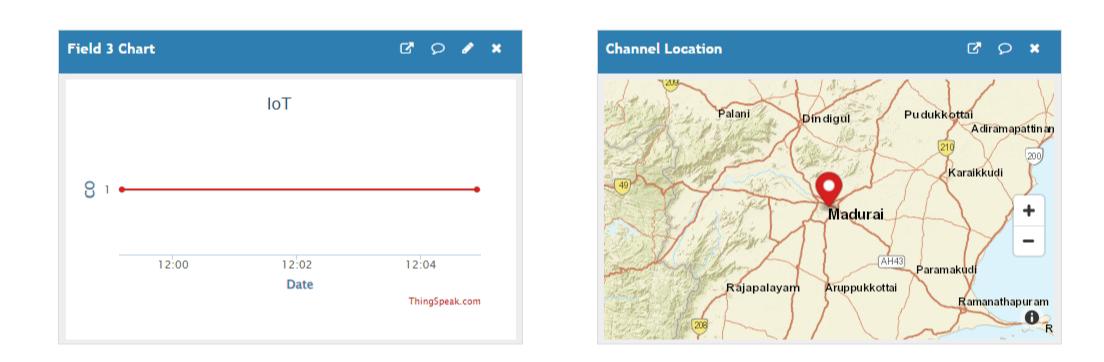
3.Note Your API Key:

In the channel settings, you’ll find an API Key. You will need This key to send data to your ThingSpeak channel

4.Integrate ThingSpeak in Your Python Code:

In your Python code running on Wokwi, use the Urequests library to send data to ThingSpeak. You can Construct a URL with your API Key and the data you want To send.

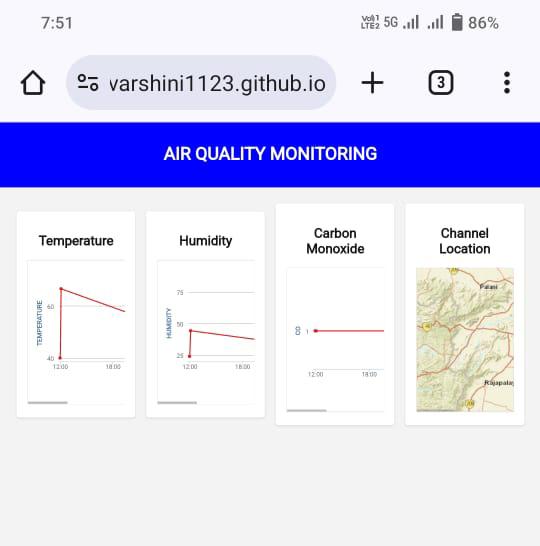




WEB PLATFORM

Our website is a user-friendly platform designed For air quality enthusiasts and data aficionados. It provides Real-time access to essential quality information, such As temperature and humidity. The site offers an intuitive and Visually appealing interface, displaying data in an easily Digestible format Users can stay informed about air quality conditions in Their preferred locations, be it environment, gardens, or other outdoor Settings. The website extracts data from ThingSpeak, ensuring Accuracy and reliability.

This data is presented with clean, organized design elements, allowing users to track Environmental trends and make informed decisions. Our website is a valuable resource for both casual observers and Serious environmentalists, about air quality offering a seamless and Experience for exploring and understanding the world around us.



YOU CAN ACCESS OUR WEBSITE USING THE BELOW URL

[https://Varshini1123.github.io.git](https://github.com/Varshini1123/Varshini1123.github.io.git)

Implications for Public Health and Environment:

IOT air quality monitoring has significant implications for public health and the environment. It empowers communities and policymakers to:

Protect Public Health: By providing accurate, real-time data, IOT air quality monitoring helps alert individuals and authorities to potential health risks, allowing for timely interventions to reduce exposure to harmful pollutants.

Support Environmental Policies: The data generated by IoT systems can inform the development of evidence-based policies aimed at reducing pollution, mitigating climate change, and promoting sustainable urban planning.

Raise Awareness: Accessible air quality information increases public awareness and encourages individuals to make informed choices, such as adjusting their daily routines to reduce exposure to pollution.

CONCLUSION



Air quality monitoring in environmental is a crucial practice that Harnesses the power of IOT and web technologies to ensure the health And sustainability of our natural environments. By simulating this Process with platforms like Wokwi, we can develop and fine-tune our Monitoring systems in a safe and controlled digital environment before Implementing them in real-world park settings. The integration of ThingSpeak further enhances our ability to collect, analyse, and display Vital data. As technology continues to advance, the energy between IOT, web technologies, and Conservation holds the promise of a more informed and interconnected Future for both the protection of our environment and the quality of outdoor enthusiasts.