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

ENVIRONMENTAL MONITORING IN PARKS
Phase_4

PROJECT MENTOR

R.M.SENTHIL KUMAR

K.RANJITH KUMAR (AU923821106041)
G.RAGHUPATHI (AU923821106040)
G.SARAVANA BALAJI (AU923821106046)
S.AZIMMALIK (AU923821106010)
T.ARUN (AU923821106006)

ENVIRONMENTAL MONITORING IN PARKS

(PHASE-4)

INTRODUCTION

Environmental monitoring in parks has taken a significant step forward with the integration of IoT technology and web solutions. This approach allows us to collect real-time data on park conditions and wildlife behavior. Through simulations, we can see how these technologies create a virtual park environment, helping us understand and protect our natural spaces more effectively.

IoT sensors placed within the parks collect data on various environmental factors, which is then sent to central systems using the Internet. Web technologies process this data and create a virtual park ecosystem, mirroring the real world. This simulation helps us predict the impact of changes and conservation efforts.

The combination of IoT and web technologies makes environmental data accessible to park rangers, scientists, and the public, enabling informed decision-making and promoting sustainable practices. This marks a significant advancement in the world of environmental monitoring and highlights the power of technology in conservation efforts.

Environmental Monitoring in Parks:

Environmental monitoring in parks is a way to keep an eye on nature. It's like checking the health of the park's environment. This means looking at things like air, water, animals, and plants. People use special tools to collect information about these things.

This information helps park managers make good decisions. They can keep the park safe for people and animals. It's also important for understanding how the environment is changing, like if the weather is getting warmer. So, environmental monitoring is like taking care of our parks so they stay healthy and enjoyable for everyone.

In a park, sensors and devices are used to measure different things. For example, they can tell us if the water in a lake is clean or if there's enough food for animals. The data collected by these devices is sent to a central place for analysis. This helps park rangers and scientists make the right choices to protect the environment and make sure visitors have a safe and enjoyable time in the park. It's all about looking after nature and keeping our parks beautiful for everyone to enjoy.

SIMULATION PROCESS

WOKWI:

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 environmental monitoring system in a park using Wokwi, you'll need to follow these steps:

1.Creating 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 environmental monitoring projects, the ESP32 is a popular choice due to its built-in Wi-Fi capabilities.

3.Design Circuit:

- Using Wokwi's intuitive drag-and-drop interface, design the circuit for your environmental monitoring system. This may include adding sensors (e.g., DHT11 for temperature and humidity), LEDs, 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 `urequests` module to send data to a server or ThingSpeak.

5.Simulate Project:

- Click the "Run" button to simulate your project. You can observe how your Python code interacts with the virtual environment, sensors, and microcontroller.

6.Test and Debug:

- Use the simulation environment to test your code for environmental 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 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 environmental monitoring system. For example, you can add sensors for air quality, 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 ThinkSpeak, a cloud – based IoT platform .

```
# PROGRAM STARTS
import machine
import time
import network
import dht
import urequests

# DHT22 data pin connected to GPIO5 (D1)
dht_pin = machine.Pin(15)
# WiFi credentials
ssid = "Wokwi-GUEST"
password = ""
```

```
# ThingSpeak API key
api_key = "Z87Z4HE43PSFYE9W"

# ThingSpeak server
ts_server = "api.thingspeak.com"

channel_number = 2307255



# 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 and humidity
def read_dht():
    dht22 = dht.DHT22(dht_pin)  # Create the DHT22 object
    dht22.measure()
    temp, hum = dht22.temperature(), dht22.humidity()

  # Read temperature and humidity
    return temp, hum



# Send data to ThingSpeak
def send_to_thingspeak(temp, hum):
    base_url = "http://{}/update?
api_key={}&field1={}&field2={}".format(ts_server, api_key, temp, hum)
    response = urequests.get(base_url)
    if response.status_code == 200:
        print("Data sent to ThingSpeak successfully")
    else:
        print("Failed to send data to ThingSpeak")
```

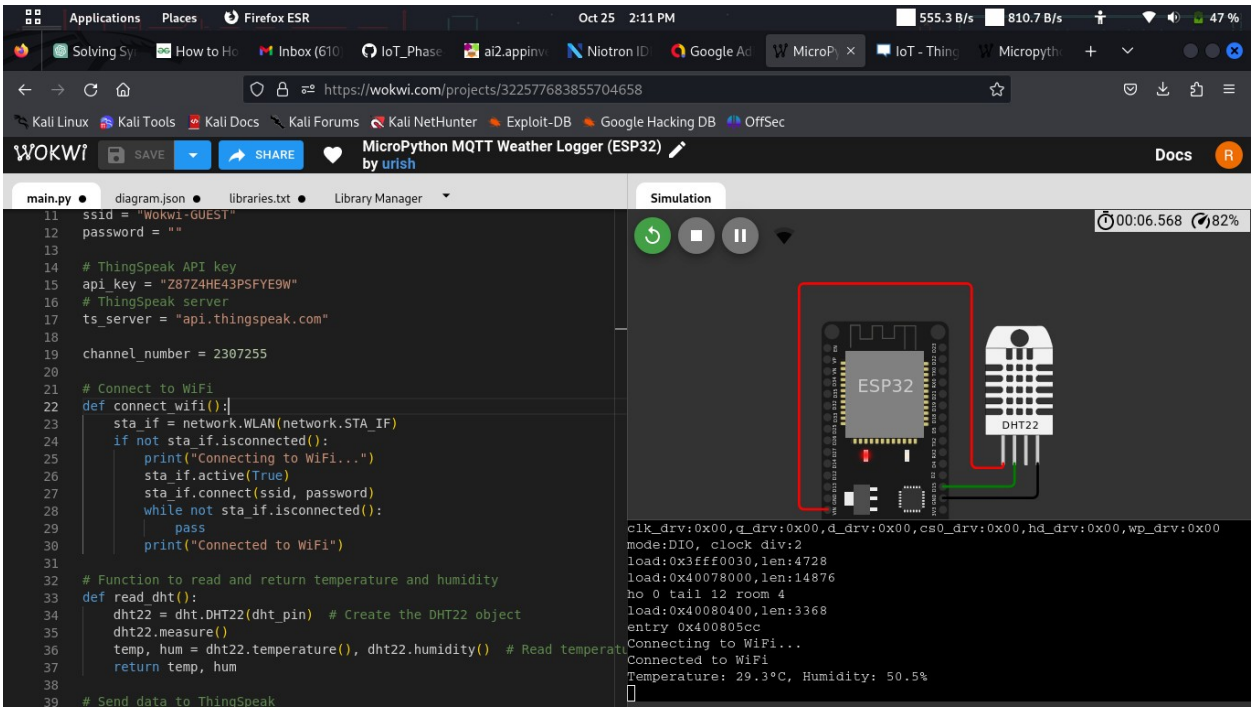
```
# Main program
def main():
    connect_wifi()
    while True:
        temp, hum = read_dht()
        print("Temperature: {}°C, Humidity: {}%".format(temp, hum))
        send_to_thingspeak(temp, hum)
        time.sleep(300)  # Send data every 5 minutes

if __name__ == '__main__':
    main()
```

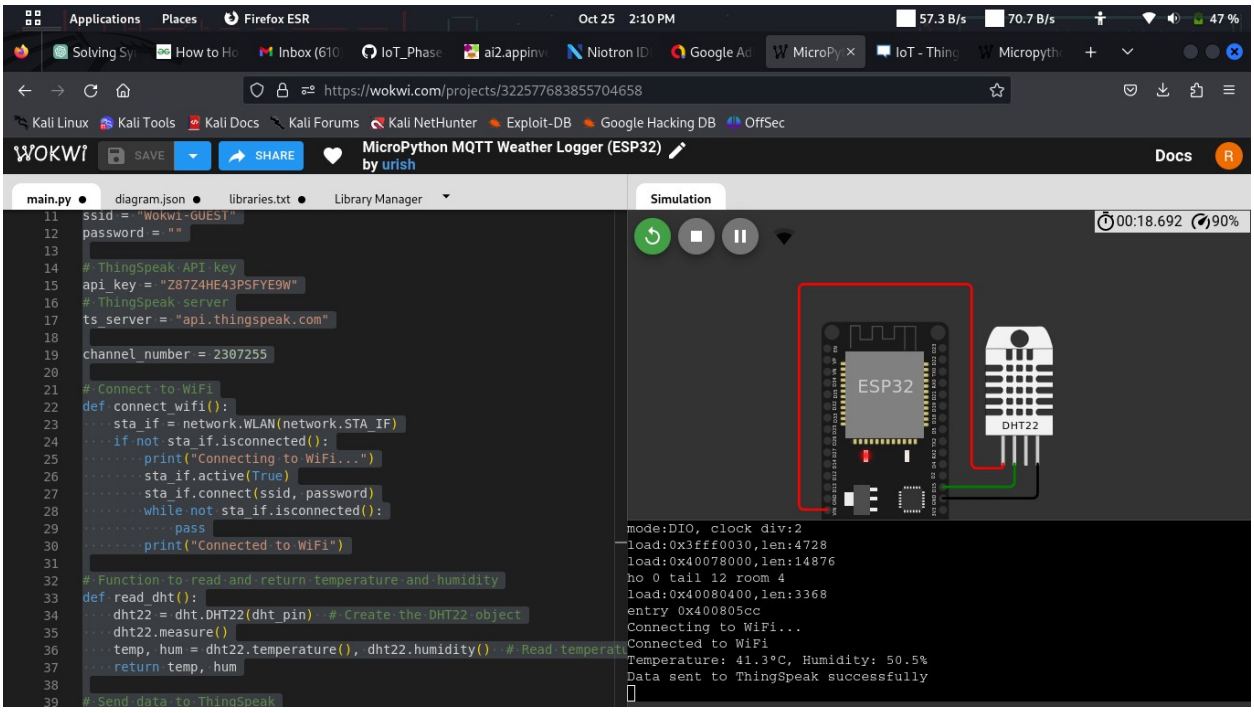
The Above Program Send the data to the thingspeak server. The Data Get From the Microcontroller ESP32.Humdity and Temperature can be Get using the DHT 22 Sensor. The Data will send for Every 5 Minutes

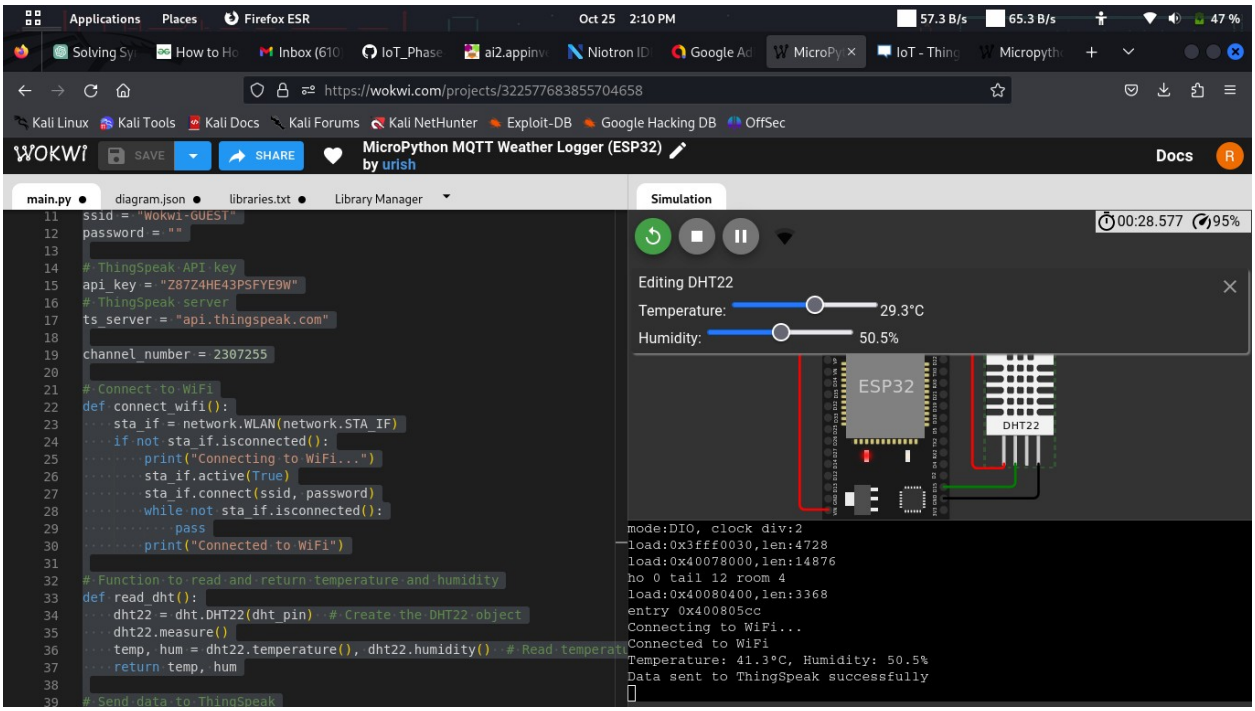
The Program Access The ThingSpeak server Via the API key

OUTPUT in WOKWI



The Two Different Outputs are shown





We can adjust the Temperature and Humdity level of the sensor because this is the simulation process not having a physical components

The Follwing Library Files are used in WOKWI

We Must include The Library Files

- DHT22
- DHT sensor library for ESPx
- WiFi
- ThingSpeak

THINGSPEAK

ThingSpeak is our chosen server for environmental monitoring project

1.Set Up ThingSpeak Account:

- If you haven't already, create an account on ThingSpeak (https://thingspeak.com/).

2.Creating a ThingSpeak Channel:

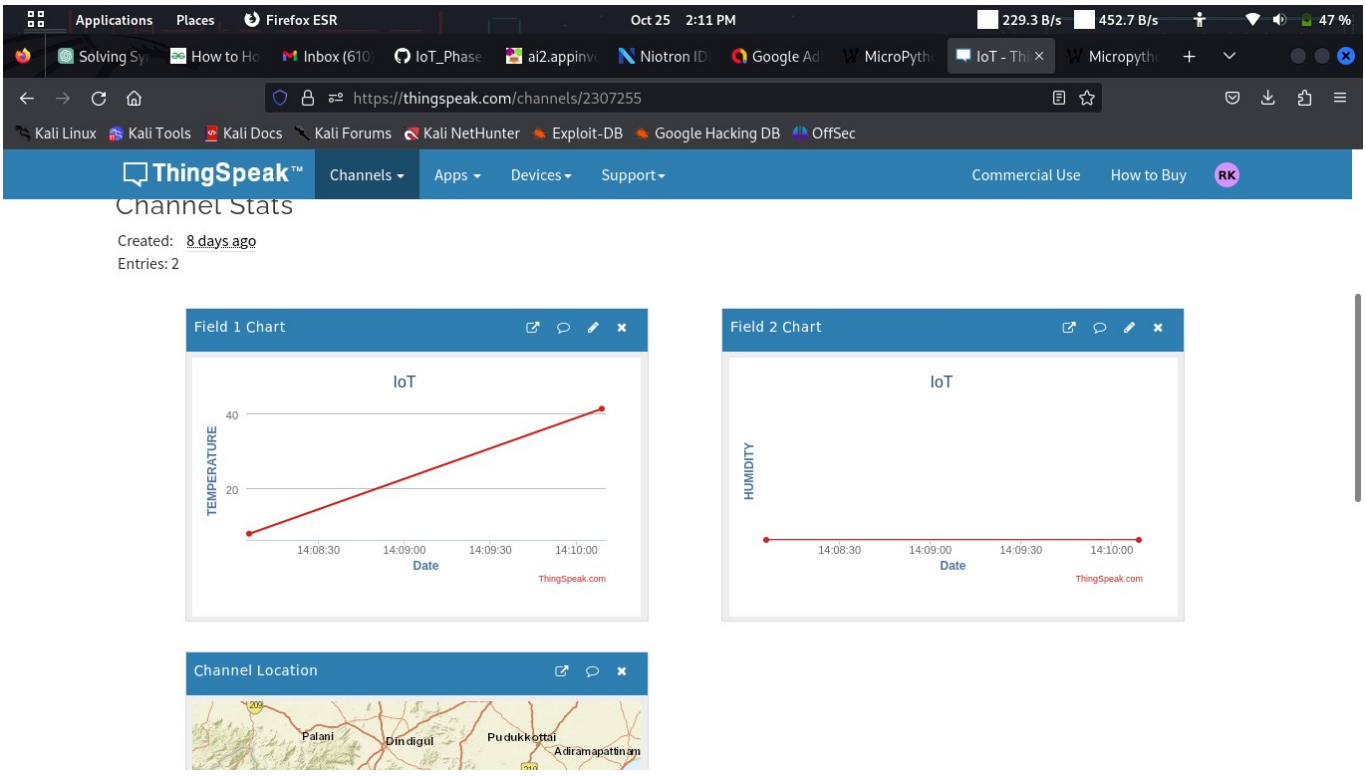
- In your ThingSpeak account, create a new channel. This channel will be used to store the data from your environmental monitoring system.

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:

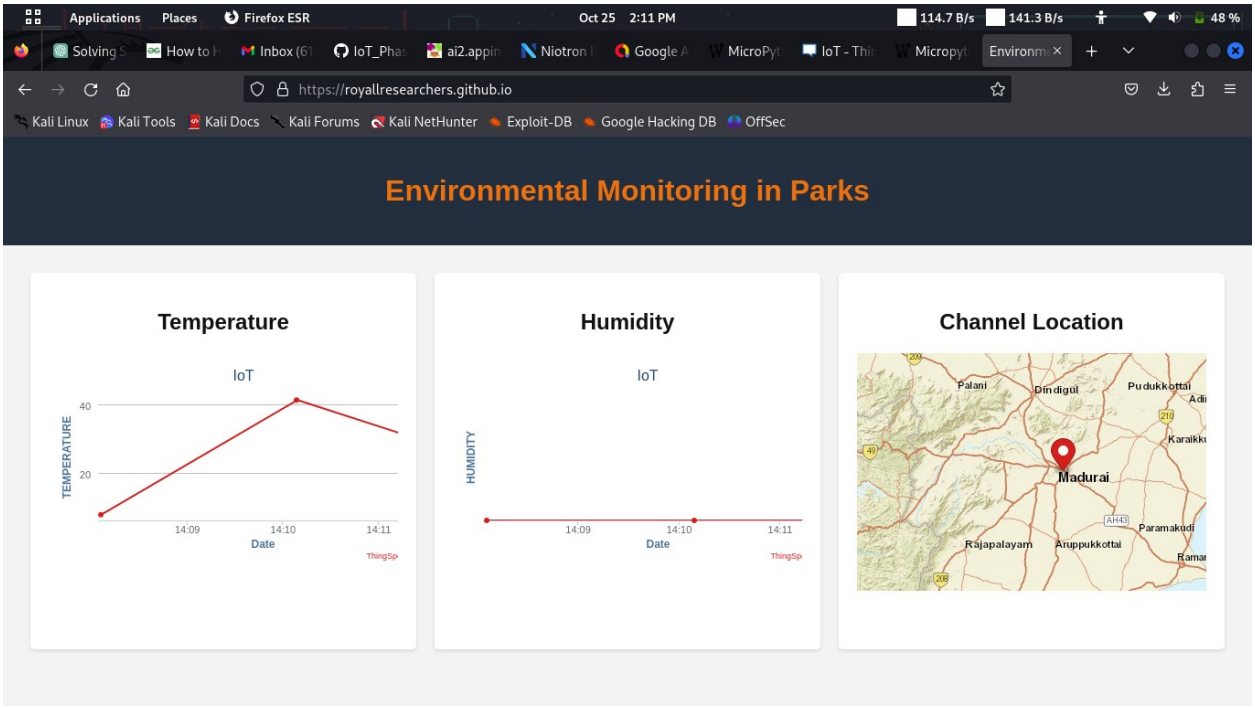
- In your Python code running on Wokwi, use the `urequests` library to send data to ThingSpeak. You can construct a URL with your API Key and the data you want to send.



WEB PLATFORM

Our website is a user-friendly platform designed for environmental enthusiasts and data aficionados. It provides real-time access to essential environmental 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 environmental conditions in their preferred locations, be it parks, 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, offering a seamless and enjoyable experience for exploring and understanding the world around us.

YOU CAN ACCESS OUR WEBSITE USING THE BELOW URL

https://royallresearchers.github.io

our website is hosted on Github

CONCLUSION

environmental monitoring in parks 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, analyze, and display vital environmental data. As technology continues to advance, the synergy between IoT, web technologies, and environmental conservation holds the promise of a more informed and interconnected future for both the protection of our parks and the enjoyment of outdoor enthusiasts.