Environmental Monitoring Using IoT

1.For a environmental monitoring project using IoT,Objectives could include real-time data collection, analysis,and providing for effective urban planning or public awareness.

2.Temperature and humidity level monitoring using IoT by displaying in publice places where this IoT system deployed in specify children park ,theme park,tourist places ,this places real-time Temperature and humidity levels displyed.

3.To this project goal people choosan for convenient place such as them park,park, tourist places to goes.

To setup an IoT device,for Environmental monitoring, consider using

* a microcontroller like Arduino Uno.

* a sensor used such as DHTT11.

* a communication module like ESP8266

Connect the sensor to the microcontroller,program it to gather and process temperature and Humidity data to a central server for analysis.

SPECIFICATION:

Hardware:

1. Temperature and Humidity sensor observing the fixed level.
2. Microcontroller processing the data.
3. Communication Module transmit data to a central server.
4. Power supply: limiting the power source to avoiding the short circuit or over load.
5. Protection: protect the IoT system by causing environmental factors like rain.

Software:

1. Embedded Software for Microcontroller programs the microcontroller for data collection and transmission.

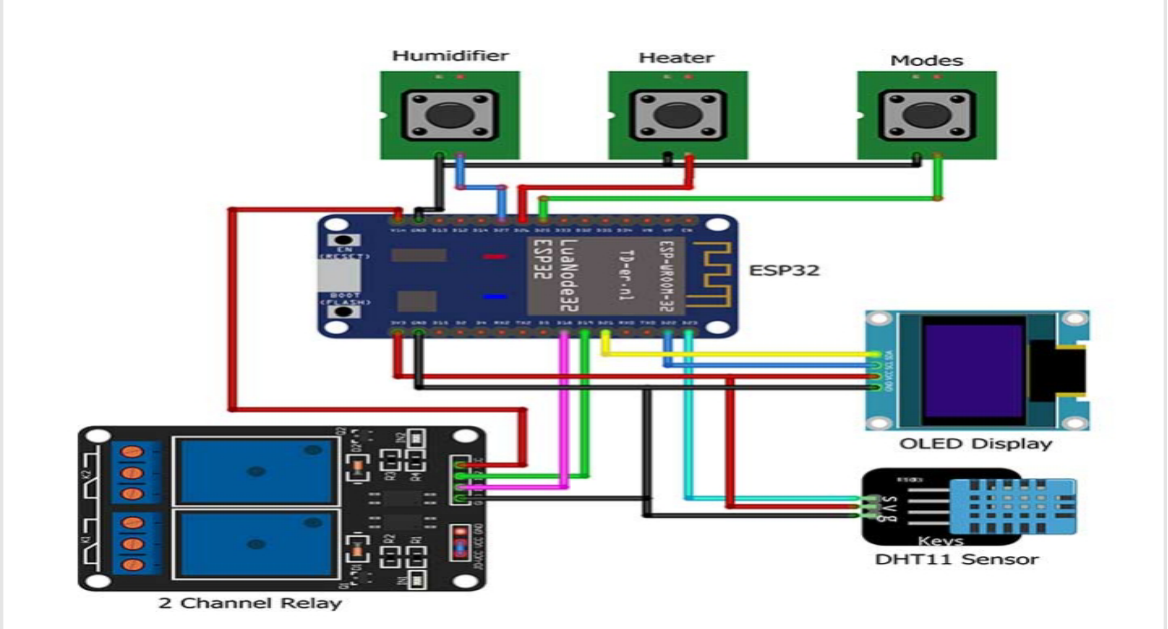
2. Server-side side software :manages and analysis.

3. Data base : stores historical noise data for analysis.

4. Web/App Interface: allowes users to acces and visualise noise pollution data

5. Data Analytics Tools : Processing and increasing temperature and humidity level for meaningfull insights.

circuitdiagram



Monitoring the temperature and humidity levels using IoT involves deploying sensor and data collection devices to gather information about temperature and humidity levels in different places.

* **Sensore Deployment:**Fixed temperature and humidity sensors at various park, tourist places

* **IoT Connection:** Connect the DHTT11 sensor to the internet using wireless technologies using wifi.This allows the sensors to transmit data.

* **Data Collection:**Collect data the data from DHTT11 sensors continuesly.This sensor slould measure temperature and Humidity at sorrounding.

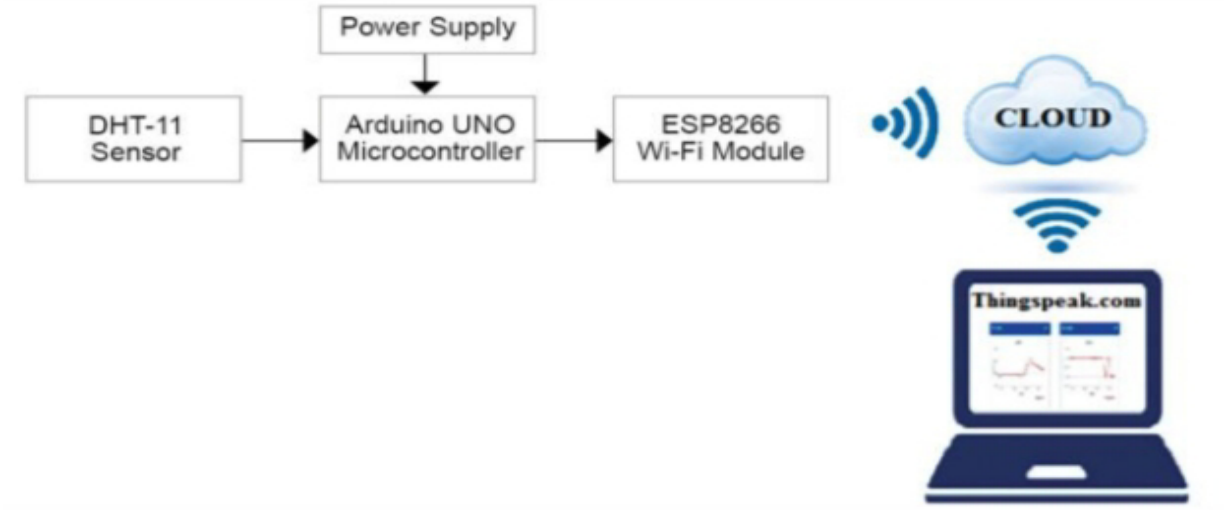
* **Data storage:** Store the collected data in a source and a secure and scalable database,cloud database like Google cloud.

***Data Analytics:** Analysis the temperature and humidity data to identify patterns and trends. You can machine learning algorithms to detect anomalies or specific noise events that may indicate levels.

***Visualization:**Create dashboards or apps to visualise the levels of data. This can help both authorities and public understand temperature and humidity levels in real-time or historically.

***Alerts and Notification:** setup automated alerts and notifications for noise levels that exceed predefined thresholds.

Block diagram



Python cod:

```
Prin("Environmental monitoring")
Import machine
Import time
import ujson
Import urequests

 Define the MQTT broker parameters
MQTT_BROKER = “localhost”
MQTT_PORT = 1883
MQTT_TOPIC = “/environment/data”

# Define the DHT22 sensor parameters
DHT22_PIN = 13

# Define the DHTT sensor objects
dhtt = machine .DHTT(DHT22_PIN)

# Connect to the MQTT broker
Client = urequests.clients( )
Client.connect(MQTT_BROKER, MQTT_PORT)
```

```
# Publish the environment data to the MQTT topice

Def publish_data(temperature, humidity);

    Data = {“temperature”: temperature, “humidity” : humidity}

    json_data = ujson.dump(data)

   client.publish(MQTT_TOPIC, json_data)

# Start a loop to read the DHT22 sensor and publish the data to the MQTT broker

While True :

        # Read the temperature and humidity from the DHT22 sensor temperature, humidity = DHT22.read() # Publish the environment data to the MQTT topic publish_data(temperature, humidity)

      # Wait for 10 seconds before reading the sensor again

Time.sleep(10)
```

HTML code:

```
<!DOCTYPE html>

<html>

<head>

  <title>DHT11 Sensor Data</title>

</head>

<body>

  <h1>DHT11 Sensor Data</h1>

  <p>Temperature: <span id="temperature">Loading...</span> &deg;C</p>
```

```
    <p>Humidity: <span id="humidity">Loading...</span> %</p>
        <script src="script.js"></script>
</body>
</html>
```

Javascript:

```
// Replace with the URL where your server will provide sensor data
const dataUrl = 'http://your-server-url/data';

// Function to update sensor data on the web page
function updateSensorData() {
    fetch(dataUrl)
        .then(response => response.json())
        .then(data => {
            document.getElementById('temperature').textContent = data.temperature;
            document.getElementById('humidity').textContent = data.humidity;
        })
        .catch(error => {
            console.error("Error fetching sensor data:", error);
        });
}
```

```
// Update sensor data when the page loads
updateSensorData();

// Set a periodic update (e.g., every 5 seconds)
setInterval(updateSensorData, 5000);
```