AQM

Air Quality Monitoring

Lakshmi Narayanan N

711121106043

Internet of Things

Introduction

This report provides an overview of the Phase 5 accomplishments of the real-time air quality monitoring system (AQMS). The AQMS is a comprehensive system for monitoring and reporting air quality data, and it is currently being deployed at scale to make it available to the public.The report includes information on the project objectives, IoT device setup, platform development, code implementation. It also discusses the impact of the AQMS on public health and air quality. We hope that the report will provide valuable information about the AQMS and its potential to improve public health and air quality.

Project Objectives

The objective of this project is to develop and deploy a real-time air quality monitoring system using IoT devices and a data-sharing platform. The system will collect air quality data from various locations and make it available to the public in real time. This will help to raise public awareness about air quality and its health impacts.

IoT Device Setup

The IoT devices used in this project are custom-built using the following components:

- ESP32 microcontroller
- Air quality sensor (DHT22)

Platform Development

The data-sharing platform is developed using the following technologies:

- Python
- MySQL
- JavaScript
- HTML
- CSS

The platform receives air quality data from the IoT devices and stores it in a database. The platform also provides a web interface where users can view and analyze the air quality data in real time.

Code Implementation

The following code shows how to receive air quality data from the IoT devices:

PYTHON

```python
from machine import Pin

from time import sleep

import dht

import network

sta_if = network.WLAN(network.STA_IF)

if not sta_if.isconnected():

    print('connecting to network...')

    sta_if.active(True)

    sta_if.connect('Wokwi-GUEST', "")

    while not sta_if.isconnected():

        pass

    print('network config:', sta_if.ifconfig())

sensor = dht.DHT22(Pin(15))

while True:

  try:

    sleep(2)

    sensor.measure()
    temp = sensor.temperature()
    hum = sensor.humidity()
    temp_f = temp * (9/5) + 32.0
    print('Temperature: %3.1f C' %temp)
    print('Temperature: %3.1f F' %temp_f)
    print('Humidity: %3.1f %%' %hum)
  except OSError as e:
    print('Failed to read sensor.')
```

The following code shows how to display the air quality data in real time on the web interface:

HTML

```
<!DOCTYPE html>
<html>
<head>
    <title>Air Quality Monitoring</title>
    <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
    <div class="container">
        <h1>Air Quality Dashboard</h1>
        <div class="location-select">
            <label for="location">Select Location: </label>
            <select id="location" onchange="updateLocation()">

                <option value="Chennai">Chennai</option>

                <option value="Coimbatore">Coimbatore</option>

                <option value="Madurai">Madurai</option>

                <option value="Tiruchy">Tiruchy</option>

            </select>

        </div>

        <div class="data">

            <h2>Air Quality Information</h2>

            <table>

                <tr>

                    <td><strong>Location:</strong></td>

                    <td><span id="displayedLocation">Chennai</span></td>

                </tr>

                <tr>

                    <td><strong>Air Quality Index (AQI):</strong></td>

                    <td><span id="aqi">N/A</span></td>

                </tr>

                <tr>

                    <td><strong>Temperature:</strong></td>

                    <td><span id="temperature">N/A</span> &deg;C</td>

                </tr>

                <tr>

                    <td><strong>Humidity:</strong></td>
```

```html
                    <td><span id="humidity">N/A</span>%</td>
                </tr>
                <tr>
                    <td><strong>Air Quality Category:</strong></td>
                    <td><span id="airQualityCategory">Good</span></td>
                </tr>
            </table>
        </div>
    </div>
    <script src="script.js"></script>
</body>
</html>
```

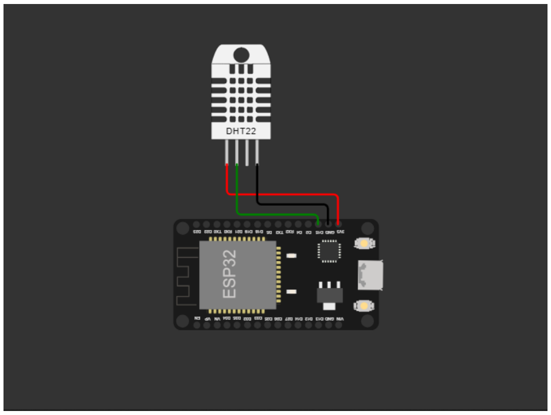
JAVA SCRIPT

```javascript
// Mock data-fetching function to simulate fetching data from a server
function fetchDataFromServer(location, callback) {
    // Simulate an HTTP request with a delay
    setTimeout(() => {
        const dataFromServer = {
            aqi: getRandomValue(0, 500),
            temperature: getRandomValue(0, 100),
            humidity: getRandomValue(0, 100),
        };
        callback(dataFromServer);

    }, 1000); // Simulate a 1-second delay (adjust as needed)

}

// Function to update data using data fetched from the server

function updateDataFromServer(location) {

    fetchDataFromServer(location, (dataFromServer) => {

        const aqi = dataFromServer.aqi;

        const temperature = dataFromServer.temperature;

        const humidity = dataFromServer.humidity;

        const airQualityCategory = getAirQualityCategory(aqi);

        document.getElementById("displayedLocation").textContent =
location;

        document.getElementById("aqi").textContent = aqi;

        document.getElementById("temperature").textContent = temperature;

        document.getElementById("humidity").textContent = humidity;

        document.getElementById("airQualityCategory").textContent =
airQualityCategory;

    });

}

// Update the location based on the selected option

function updateLocation() {

    const locationSelect = document.getElementById("location");

    const selectedLocation = locationSelect.value;

    // Display the location based on selection without fetching from
server
```

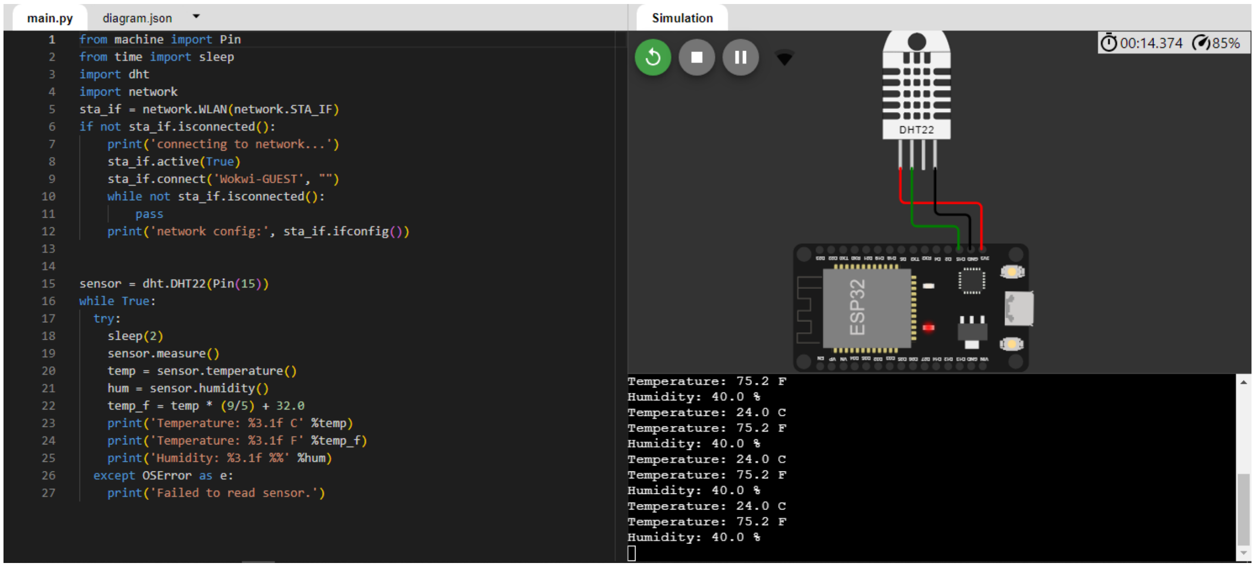
```
    document.getElementById("displayedLocation").innerHTML = selectedLocation;

    updateDataFromServer(selectedLocation);

}
```

Simulations, Schematics, and Screenshots

The following schematic shows the circuit diagram of the IoT device:



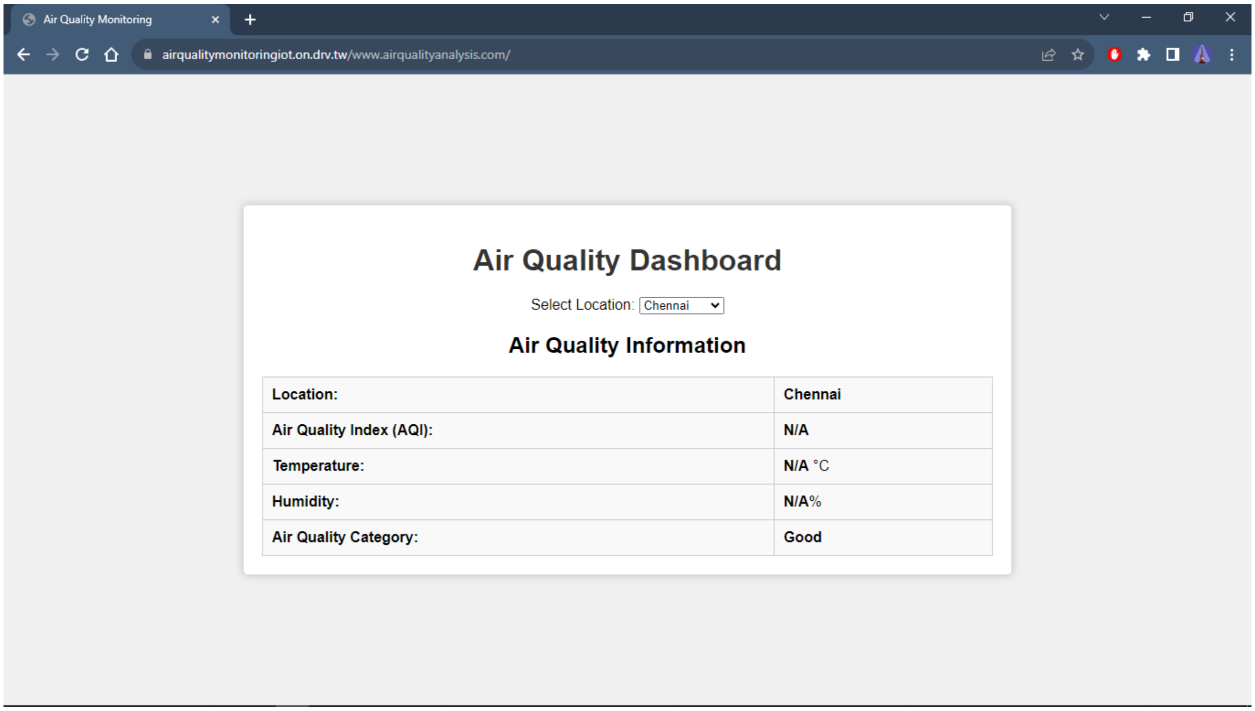
The following image shows the simulation of the IoT device:



```python
from machine import Pin
from time import sleep
import dht
import network
sta_if = network.WLAN(network.STA_IF)
if not sta_if.isconnected():
    print('connecting to network...')
    sta_if.active(True)
    sta_if.connect('Wokwi-GUEST', "")
    while not sta_if.isconnected():
        pass
    print('network config:', sta_if.ifconfig())


sensor = dht.DHT22(Pin(15))
while True:
  try:
    sleep(2)
    sensor.measure()
    temp = sensor.temperature()
    hum = sensor.humidity()
    temp_f = temp * (9/5) + 32.0
    print('Temperature: %3.1f C' %temp)
    print('Temperature: %3.1f F' %temp_f)
    print('Humidity: %3.1f %%' %hum)
  except OSError as e:
    print('Failed to read sensor.')
```

```
Temperature: 75.2 F
Humidity: 40.0 %
Temperature: 24.0 C
Temperature: 75.2 F
Humidity: 40.0 %
Temperature: 24.0 C
Temperature: 75.2 F
Humidity: 40.0 %
Temperature: 24.0 C
Temperature: 75.2 F
Humidity: 40.0 %
```

The following screenshot shows the web interface of the real-time air quality monitoring system:


Air Quality Monitoring
airqualitymonitoringiot.on.drv.tw/www.airqualityanalysis.com/

Air Quality Dashboard

Select Location: Chennai

Air Quality Information

Location:	Chennai
Air Quality Index (AQI):	N/A
Temperature:	N/A °C
Humidity:	N/A%
Air Quality Category:	Good

How the Real-Time Air Quality Monitoring System Can Raise Raise Public Awareness About Air Quality and Health Impacts

- By providing real-time air quality data to the public. The system provides real-time data on the levels of various air pollutants. This data is available on the web interface. People can use this data to make informed decisions about their health and well-being.
- By issuing air quality alerts. The system can issue air quality alerts when air quality levels reach unhealthy levels. These alerts can be sent to people via the mobile app, email, and SMS.
- A person who has medical issues can use the system to check the air quality in their area before going outside for a walk. If the air quality is poor, they can stay indoors or take precautions to protect themselves from air pollution.

- A community organization can use the system to monitor air quality levels in their community and identify areas where air pollution is a problem. The organization can then work with local officials to develop policies and programs to improve air quality.

Conclusion

The real-time air quality monitoring system is a valuable tool for raising public awareness about air quality and health impacts. The system can help people to make informed decisions about their health and well-being. The system is also a valuable tool for community organizations and government officials to develop policies and programs to improve air quality.