ENVIRONMENTAL MONITORING IN PARKS USING IOT WITH SENSORS

PROJECT DESCRIPTION:

This project tends to frequently monitoring the climatic changes through the deployment of IOT devices equipped with various sensors.The collected data will be processed using python scripts to improve operational efficiency,provide real time informations to visitors and enhance the overall monitoring.

SENSORS FOR DEPLOYMENT:

To monitor climatic changes,several type of sensors are used.Here are the key sensor and their applications:

ARDUINO UNO:

it is used to connect both hardware and software elements.

DHT22 SENSORS:

Applications:

It uses a capacitive humidity sensor and a thermister to measure the surrounding air and spits out a digital signal on the data pin.

Benefits:

it uses a polymer capacitor to sense the temperature and humidity measuring the teamperature of the air between -40 and 80 degrees centigrad and the realative humidity between 0 and 100 %.

BMP 180 SENSOR:

Applications:

BMP180 can read and is responsive to changes in temperature.

MQ-7(GAS SENSOR):

Applications:

it is used for domestic gas leakage detector.

Benefits:

Good sensitivity to combustible gas in wide range.

PROJECT STEPS:

Connect Sensors: Wire up the sensors to your Arduino Uno using the appropriate pins. Ensure you have the necessary libraries installed to interface with the sensors.

Programming: Write Arduino code to read data from the sensors and send it to a server or IoT platform. You might use libraries like "Adafruit\_IO" or "ThingSpeak" for easy data transmission.

Internet Connectivity: If you're using a Wi-Fi module, set up the connection to your local Wi-Fi network. If you're using an Ethernet shield, connect it to your network using an Ethernet cable.

Data Transmission: Periodically collect data from the sensors and send it to a server. You can transmit data using HTTP, MQTT, or other IoT protocols.

Data Visualization: Use a platform like Adafruit IO, ThingSpeak, or a custom web application to display and analyze the collected data. You can create graphs, alerts, and more.

Power Supply: Make sure you have a stable power supply, especially if you plan to run the project continuously. You might consider a battery pack or a power adapter.

Housing: Depending on where you want to monitor the environment, consider a weatherproof housing if needed.

Testing and Calibration: Test your setup and ensure your sensors are correctly calibrated for accurate readings.

Monitoring: Keep an eye on your system's performance and data accuracy. You might need to make adjustments or improvements over time.

Security: If the data is sensitive, ensure data security and encryption during transmission.

PYTHON PROGRAM:

python

Copy code

import dht

import machine

import time

import urequests

# Configure your Wi-Fi credentials

WIFI\_SSID = "Your\_SSID"

WIFI\_PASSWORD = "Your\_Password"

# ThingSpeak API endpoint and API Key

THINGSPEAK\_API\_KEY = "Your\_API\_Key"

THINGSPEAK\_URL = "https://api.thingspeak.com/update"

# Create a DHT22 sensor object

dht\_sensor = dht.DHT22(machine.Pin(4))

# Connect to Wi-Fi

import network

wifi = network.WLAN(network.STA\_IF)

wifi.active(True)

wifi.connect(WIFI\_SSID, WIFI\_PASSWORD)

# Function to read sensor data and send to ThingSpeak

def send\_data\_to\_thingspeak(temperature, humidity):

data = "field1={:.2f}&field2={:.2f}".format(temperature, humidity)

response = urequests.get(THINGSPEAK\_URL, params=data, headers={"api\_key": THINGSPEAK\_API\_KEY})

print("Response:", response.text)

response.close()

while True:

try:

# Read sensor data

dht\_sensor.measure()

temperature = dht\_sensor.temperature()

humidity = dht\_sensor.humidity()

print("Temperature: {:.2f}°C, Humidity: {:.2f}%".format(temperature, humidity))

# Send data to ThingSpeak

send\_data\_to\_thingspeak(temperature, humidity)

except Exception as e:

print("Error:", e)

# Delay between readings (e.g., every 15 minutes)

time.sleep(900)