ENVIRONMENTAL MONITORING IN PARKS

Phase 3

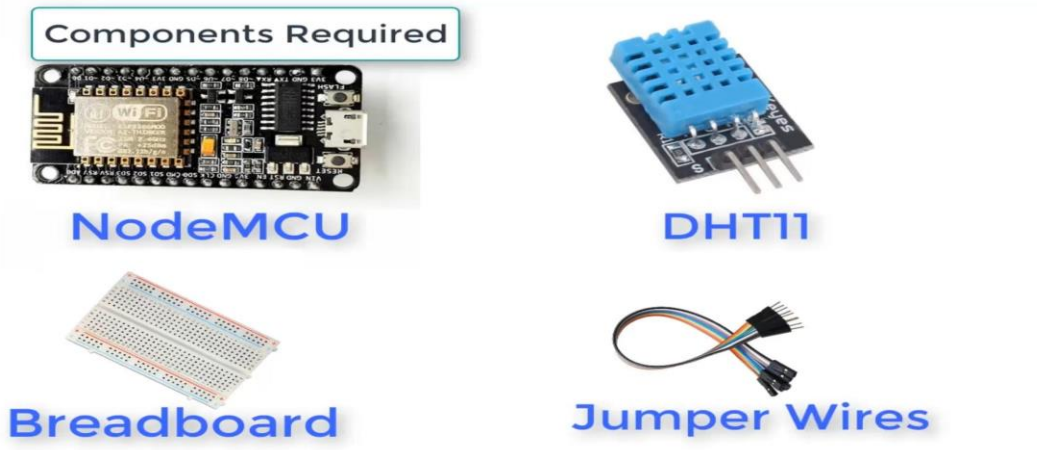
TEAM MEMBERS:

UDHAYAKUMAR N	(810021106090)
NISHANTHAN T	(810021106053)
SURYAKUMAR R	(810021106308)
SANTHOSH T	(810021106067)
TARUN MR	(810021106086)
VIMAL AS	(810021106097)

Introduction:

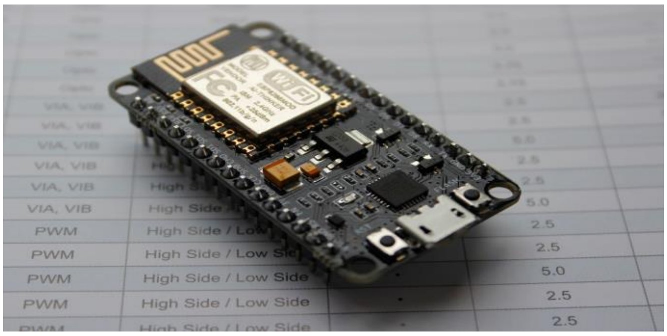
In this project, we'll use DHT11 to transmit data from temperature and humidity sensors to Thingspeak. Using the ThingSpeak IoT server, we can use this way to monitor the temperature and humidity data from our DHT11 sensor over the internet. The logged data and time graph can be viewed on the ThingSpeak dashboard. NodeMCU retrieves the current humidity and temperature from DHT11 and delivers them to the ThingSpeak server for global live monitoring.

Components Required:



- NodeMCU ESP8266
- DHT11 Temperature and Humidity Sensor
- Jumper Wires
- Breadboard

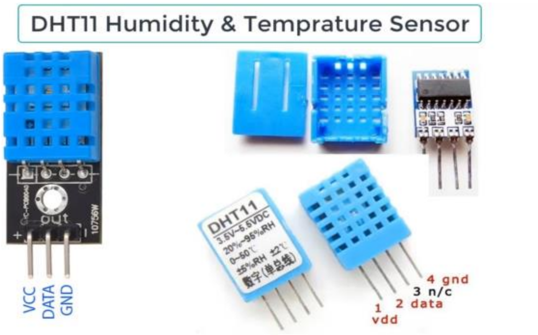
NodeMCU:



nodeMCU ESP8266

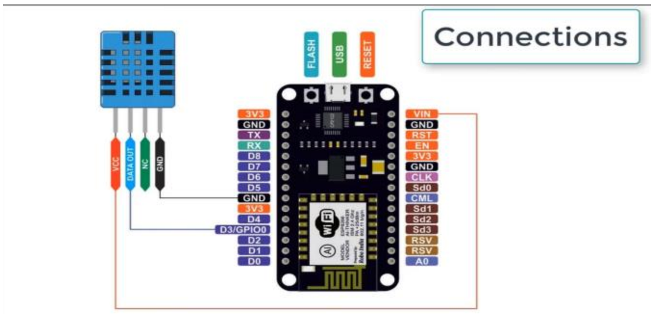
NodeMCU is an open-source firmware and development board that is based on the ESP8266 WiFi module. It allows for easy development of Internet of Things (IoT) and other embedded projects. NodeMCU provides a firmware that runs on the ESP8266, making it easy to write code in Lua or use the Arduino IDE to program the board. It's popular for its affordability and built-in WiFi capabilities, which make it suitable for a wide range of IoT applications.

DHT11 Temperature and Humidity Sensor:



The DHT11 Temperature and Humidity Sensor is an inexpensive digital sensor that measures temperature (0°C to 50°C) and humidity (20% to 80%) with modest accuracy. It provides data in digital form, making it easy to use with microcontrollers, and it's commonly employed in DIY projects and basic environmental monitoring due to its affordability and simplicity. However, it may not be suitable for applications demanding high precision or extreme environmental conditions.

Circuit Connections:



- ➔ First, the GND of the DHT11 sensor is connected to the GND of the NodeMCU.

- ➔ Second, the VCC of the DHT11 sensor is connected to the VIN of the NodeMCU.

- ➔ Finally, DATA OUT terminal of the DHT11 sensor is connected to the D3 terminal of the NodeMCU.

Characteristics:

1. Real-Time Data: Continuous monitoring and real-time data collection to provide up-to-the-minute information on environmental conditions.

2. Sensor Integration: Utilization of various sensors for temperature, humidity, air quality, light intensity, noise, soil moisture, water quality, and more.

3. Wireless Connectivity: Use of wireless communication technologies (e.g., Wi-Fi, LoRa, or cellular) for sensor data transmission to a central control system.

4. Data Visualization: Data is presented through intuitive and interactive interfaces, such as web dashboards and mobile apps for easy interpretation.

5. Historical Data Logging: Storage of historical data for trend analysis, reporting, and future planning.

6. Automated Control: Automation of environmental systems like irrigation and lighting based on sensor data.

7. Alerts and Notifications: Automatic alerts and notifications for park authorities and visitors in the event of adverse conditions.

8. Energy Efficiency: Implementation of energy-efficient technologies to reduce power consumption and environmental impact.

9. Remote Monitoring: The ability to access data and control systems remotely, enabling park management from a central location.

10. Scalability: Flexibility to add or modify sensors and components as needed to adapt to changing park requirements.

11. Sustainability: The project promotes sustainability through water conservation, energy efficiency, and reduced environmental impact.

12. Interoperability: Compatibility with various sensor types and the ability to integrate with existing infrastructure and data sources.

13. Visitor Engagement: Providing park visitors with access to environmental data and educational information through interactive displays or mobile apps.

14. Compliance and Regulation: Ensuring that the project adheres to environmental regulations and standards.

15. Maintenance and Support: Ongoing maintenance, updates, and technical support to ensure the system's reliability.

16. Customization: The system can be tailored to the specific needs and conditions of the park, allowing for unique configurations.

17. Cost-Effectiveness: Balancing functionality with cost to make the project economically viable.

18. Community Involvement: Involving the local community and park visitors in the project, fostering environmental awareness and cooperation.

19. Data Analysis and Insights: Utilizing data analytics to derive meaningful insights for better park management and decision-making.

These characteristics collectively contribute to the effective and efficient monitoring of environmental conditions in parks, enhancing visitor experiences and promoting sustainable park management.

Python Code for the NodeMCU:

```
import Adafruit_DHT
import requests

apiKey = "Your API KEY"
ssid = "WiFi Name"
```

```
password = "WiFi Password"
server = "api.thingspeak.com"
DHTPIN = 4

dht = Adafruit_DHT.DHT11

def setup():
    print("Connecting to ")
    print(ssid)
    WiFi.begin(ssid, password)
    while WiFi.status() != WL_CONNECTED:
        delay(550)
        print(".")
    print("")
    print("WiFi connected")

def loop():
    h, t = Adafruit_DHT.read_retry(dht, DHTPIN)
    if math.isnan(h) or math.isnan(t):
        print("Failed to read from DHT sensor!")
        return
    if client.connect(server, 80):
        postStr = apiKey
        postStr += "&field1="
        postStr += str(t)
        postStr += "&field2="
        postStr += str(h)
        postStr += "\r\n\r\n"
        client.print("POST /update HTTP/1.1\r\n")
        client.print("Host: api.thingspeak.com\r\n")
        client.print("Connection: close\r\n")
        client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\r\n")
        client.print("Content-Type: application/x-www-form-urlencoded\r\n")
        client.print("Content-Length: ")
        client.print(postStr.length())
        client.print("\r\n\r\n")
        client.print(postStr)
        client.stop()
    print("Data sent!")
```

Note: This code is a prototype. The vital information for the variables apiKey, ssid, password, etc. will be provided later.

Conclusion:

This project demonstrates how to utilize the DHT11 Temperature and Humidity Sensor in combination with NodeMCU to send real-time temperature and humidity data to the ThingSpeak IoT server. By doing so, it enables remote monitoring of environmental conditions over the internet, allowing users to access and visualize this data on the ThingSpeak dashboard. This project provides a practical example of how IoT technology can be employed for global environmental data tracking and analysis.