

Humidity and Temperature Data Analysis Monitoring System

Fareen Khan | Roll No: 230BCA219 | BCA 6th Semester

Abstract

This research paper presents a smart system for monitoring temperature and humidity using IoT technology. The system uses DHT sensor and ESP8266 microcontroller to collect real-time environmental data. The collected data can be analyzed using Python for better understanding and decision-making. This system is useful in agriculture, weather monitoring, and smart homes.

Introduction

Temperature and humidity are important environmental parameters. Monitoring them helps in maintaining healthy conditions. Traditional systems are manual and not accurate. This project uses IoT to automate monitoring and data analysis.

Literature Review

Many researchers have worked on IoT-based monitoring systems. Studies show that sensors like DHT11 and DHT22 are cost-effective. ESP8266 provides wireless communication. Python is widely used for data analysis and visualization.

Methodology

The system consists of DHT11 sensor, ESP8266 NodeMCU, and USB power supply. The sensor collects data and sends it to the microcontroller. The microcontroller processes and sends data to system/cloud.

Hardware Components

1. DHT11 Sensor: Measures temperature and humidity. 2. ESP8266 NodeMCU: Microcontroller with WiFi. 3. USB Cable: Power supply. 4. Jumper Wires: Connections.

Working Principle

The DHT11 sensor senses temperature and humidity. The ESP8266 reads the data and sends it to serial monitor or cloud. Python is used to analyze and visualize data.

Python Code

```
import serial
import matplotlib.pyplot as plt
ser = serial.Serial('COM3', 9600)
temp = []
hum = []
for i in range(10):
    data = ser.readline().decode()
    t, h = data.split(',')
    temp.append(float(t))
    hum.append(float(h))
plt.plot(temp)
plt.plot(hum)
plt.show()
```

Arduino Code

```
#include <DHT.h>
#define DHTPIN D4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

void setup() {
    Serial.begin(9600);
    dht.begin();
}

void loop() {
    float t = dht.readTemperature();
    float h = dht.readHumidity();
    Serial.print(t);
    Serial.print(',');
    Serial.println(h);
    delay(2000);
}
```

Results

The system provides accurate readings with low cost and power consumption. Data can be visualized using graphs.

Conclusion

This project is simple and effective. It can be extended using cloud storage and mobile applications.



```
#include <dht.h>

#include <LiquidCrystal.h> dht DHT;
LiquidCrystal lcd(11, 12, 5, 4, 3, 2);

#define DHT11_PIN 6

void setup(){
  Serial.begin(9600);
  lcd.begin(16, 2);
}

void loop(){
  int chk = DHT.read11(DHT11_PIN);

  Serial.print("Temperature = ");
  Serial.print(DHT.temperature);
  Serial.write(0xC2);
  Serial.write(0xB0);
  Serial.print("C ");
  Serial.print("Humidity = ");
  Serial.print(DHT.humidity);
  Serial.write(0x25);
  Serial.println("");
```

```
lcd.setCursor(0,0);  
lcd.print("Temp: ");  
lcd.print(DHT.temperature*9/5+32);  
lcd.print((char)223); lcd.print("F");  
lcd.setCursor(0,1);  
lcd.print("Humidity: ");  
lcd.print(DHT.humidity);  
lcd.print("%");  
  
delay(2000);  
}
```

