Report

On

**“Temperature and Humidity Monitoring System using Thingspeak”**

**Index**

|  |  |
| --- | --- |
| **Sr**  **No.** | **Title** |
| **1** | Introduction |
| **2** | Working |
| **3** | Advantages and drawback |
| **4** | Software used |
| **5** | Code and output |
| **6** | Conclusion |

# Introduction:

What is Temperature and Humidity Monitoring System ?

A Temperature and Humidity Monitoring System is a technology used to measure, track, and manage environmental conditions related to temperature and humidity levels in various settings such as buildings, warehouses, laboratories, greenhouses, server rooms, museums, and more.

This system typically consists of sensors, data loggers, and software platforms to collect, analyze, and visualize data in real-time or over specific intervals. Here's how it generally works:

1. Sensors: These are devices that measure temperature and humidity levels. They are often placed strategically throughout the area being monitored to capture accurate data. Sensors can vary in type and sophistication, ranging from basic analog sensors to more advanced digital ones.

2. Data Loggers: Data loggers are devices that collect data from the sensors at regular intervals. They store this data locally or transmit it wirelessly to a central monitoring system for analysis. Data loggers may also have additional features such as alarm notifications for when temperature or humidity levels exceed preset thresholds.

3. Software Platform: The software platform is where the data collected by the sensors and data loggers is processed, analyzed, and displayed. It provides users with real-time monitoring capabilities, historical data analysis, customizable alerts, and reports. Some systems may also offer remote access through web-based or mobile applications.

4. Alerts and Notifications: Many temperature and humidity monitoring systems are equipped with alerting mechanisms to notify users when conditions deviate from acceptable ranges. These alerts can be sent via email, SMS, or app notifications, allowing for timely intervention to prevent damage or loss.

Benefits of Temperature and Humidity Monitoring Systems include:

• Preservation of Goods: Maintaining optimal temperature and humidity levels is crucial for preserving goods such as food, pharmaceuticals, and sensitive materials.

• Compliance: Many industries are subject to regulations regarding temperature and humidity control. Monitoring systems help ensure compliance with these regulations by providing accurate records of environmental conditions.

• Cost Savings: By identifying and addressing issues such as inefficient HVAC systems or air leaks, monitoring systems can help reduce energy costs.

• Risk Mitigation: Early detection of temperature or humidity fluctuations can prevent equipment failure, product spoilage, or other costly incidents.

ESP8266:

The ESP8266 is a highly popular and versatile microcontroller module designed for Internet of Things (IoT) applications. It's known for its low cost, low power consumption, and built-in Wi-Fi connectivity, making it an ideal choice for projects requiring wireless communication and remote control capabilities. Here are some key features and characteristics of the ESP8266:

Wi-Fi Connectivity: One of the standout features of the ESP8266 is its integrated Wi-Fi module, which allows it to connect to Wi-Fi networks and communicate over the internet. This capability enables remote monitoring, control, and data transmission for IoT applications.

Microcontroller Core: The ESP8266 is powered by a 32-bit Tensilica Xtensa LX106 microcontroller core, which provides sufficient processing power for a wide range of IoT tasks.

GPIO Pins: The ESP8266 module typically comes with several general-purpose input/output (GPIO) pins that can be used to interface with sensors, actuators, and other peripheral devices. These pins can be configured for digital input/output, analog input, PWM output, and more.

Flash Memory: The ESP8266 includes built-in flash memory for program storage. Depending on the specific variant, it may have anywhere from a few hundred kilobytes to several megabytes of flash memory available for storing firmware, configuration data, and other files.

Development Environment: The ESP8266 can be programmed using a variety of development environments, including the Arduino IDE, Espressif's official ESP-IDF (IoT Development Framework), and other third-party tools and libraries. This flexibility makes it accessible to both beginners and experienced developers.

Low Power Consumption: Despite its powerful features, the ESP8266 is designed to operate with low power consumption, making it suitable for battery-powered or energy-efficient applications.

DHT11 Sensor:

The DHT11 sensor is a basic and inexpensive digital temperature and humidity sensor commonly used in electronics projects, particularly in the realm of Arduino and other microcontroller-based projects. Here are some key features and characteristics of the DHT11 sensor:

Temperature and Humidity Sensing: The primary function of the DHT11 sensor is to measure both temperature and humidity levels in the surrounding environment. It can provide temperature readings with an accuracy of ±2°C and humidity readings with an accuracy of ±5%.

Digital Output: The DHT11 sensor communicates with microcontrollers using a digital protocol, making it easy to interface with popular platforms like Arduino. It sends data as a serial signal, which can be read by the microcontroller to obtain temperature and humidity values.

Single-Wire Communication: The DHT11 sensor utilizes a single-wire communication protocol, which means it requires only one data pin for both sending and receiving data. This simplicity makes it easy to integrate into projects with limited available pins.

Relatively Simple to Use: The DHT11 sensor is relatively simple to use, especially when compared to more advanced sensors with additional features and capabilities. It typically requires minimal external components and can be connected directly to a microcontroller without the need for additional circuitry.

Operating Voltage: The DHT11 sensor operates at a voltage range of 3.3V to 5V, making it compatible with a wide range of microcontroller platforms.

**Working of Temperature and Humidity Monitoring System using ESP8266 and DHT11**

.

A Temperature and Humidity Monitoring System using ESP8266 and DHT11 involves integrating the ESP8266 microcontroller board with the DHT11 sensor to measure temperature and humidity levels. The ESP8266 is a low-cost Wi-Fi microcontroller with built-in Wi-Fi capability, making it suitable for IoT (Internet of Things) applications. The DHT11 sensor is a basic digital sensor capable of measuring temperature and humidity.

Hardware Setup:

Connect the DHT11 sensor to the ESP8266 microcontroller board. The DHT11 sensor typically has three pins: VCC, data, and ground. Connect VCC to a 3.3V or 5V pin on the ESP8266 (check the specifications of your specific board), connect ground to a ground pin, and connect the data pin to any GPIO pin on the ESP8266 (e.g., GPIO2).

Power up the ESP8266 board.

**Programming the ESP8266:**

Write code to read data from the DHT11 sensor using the GPIO pin connected to the data pin of the sensor.

Use the ESP8266's Wi-Fi capabilities to establish a connection to your local Wi-Fi network.

Implement code to send the temperature and humidity data to a server or cloud platform. You can use protocols like MQTT, HTTP, or TCP/IP for communication.

Optionally, implement error handling and data formatting to ensure reliable data transmission.

Data Transmission:

Once the ESP8266 is programmed and connected to Wi-Fi, it will start reading temperature and humidity data from the DHT11 sensor at regular intervals.

The ESP8266 will then transmit this data to the designated server or cloud platform over the Wi-Fi network.

Depending on your implementation, you may choose to transmit data continuously or trigger transmissions based on predefined conditions.

Data Processing and Visualization:

On the server or cloud platform, receive and process the incoming temperature and humidity data.

Store the data in a database for historical analysis and visualization.

Implement a user interface Thingspeak to display real-time and historical temperature and humidity data in a human-readable format.

Optionally, set up alerts or notifications to notify users of critical temperature or humidity conditions.

Monitoring and Maintenance:

Monitor the system regularly to ensure proper functioning of the hardware and software components.

Address any issues that arise, such as connectivity problems or sensor failures.

Update the system as needed to add new features or improve performance.

**Advantages:**

The Temperature and Humidity Monitoring System using ESP8266 and DHT11 has numerous practical applications across various domains. Here are some common applications:

1. Home Automation:

Monitor indoor temperature and humidity levels to ensure comfort and well-being.

Automatically control heating, ventilation, and air conditioning (HVAC) systems based on environmental conditions.

Detect and prevent issues such as mold growth or excess humidity in basements and bathrooms.

1. Greenhouse Monitoring:

Monitor temperature and humidity levels in a greenhouse to create optimal growing conditions for plants.

Automatically control ventilation, watering systems, and shading based on environmental parameters.

Receive alerts for temperature or humidity extremes that could damage crops.

1. Food Storage and Preservation:

Monitor temperature and humidity in refrigerators, freezers, and food storage areas to ensure food safety and quality.

Receive alerts for temperature fluctuations that could lead to spoilage or contamination.

Maintain optimal storage conditions for perishable goods such as fruits, vegetables, and dairy products.

1. Industrial Monitoring:

Monitor temperature and humidity levels in warehouses, manufacturing facilities, and storage areas to protect sensitive equipment and materials.

Ensure compliance with regulatory requirements for environmental conditions in industries such as pharmaceuticals and food processing.

Detect and mitigate issues such as condensation or corrosion in industrial environments.

1. Server Room Monitoring:

Monitor temperature and humidity in server rooms and data centers to prevent overheating and equipment failures.

Automatically adjust cooling systems and airflow based on environmental conditions to maintain optimal operating conditions for servers and networking equipment.

Receive alerts for temperature spikes or humidity fluctuations that could lead to downtime or data loss.

1. Environmental Monitoring:

Monitor temperature and humidity levels in outdoor environments for research, conservation, or weather monitoring purposes.

**Drawbacks**:

1. Limited Accuracy: The DHT11 sensor has relatively low accuracy compared to more expensive sensors. It has a temperature accuracy of ±2°C and humidity accuracy of ±5%. This may not be suitable for applications requiring precise measurements.
2. Limited Range: The ESP8266's WiFi range might be limited, especially in environments with many obstacles or interference. This can affect the reliability of data transmission, especially in large or obstructed spaces.
3. Power Consumption: The ESP8266 can consume a significant amount of power, especially when transmitting data over WiFi. This can be a concern for battery-powered applications, as it may reduce battery life.
4. Single Point of Failure: Since both temperature and humidity measurements rely on a single sensor (DHT11), a failure in the sensor could result in inaccurate or no data being recorded for both parameters.
5. Sensor Drift: Over time, sensors like the DHT11 may experience drift, leading to gradual inaccuracies in measurements. Regular calibration may be necessary to maintain accuracy.
6. Limited Compatibility: The ESP8266 platform and DHT11 sensor may not be compatible with all development environments or software libraries, limiting flexibility in integration with other systems or platforms.
7. Susceptibility to Environmental Factors: Both the ESP8266 and DHT11 are susceptible to environmental factors such as temperature extremes, moisture, and electromagnetic interference, which can affect their performance and reliability.
8. Limited Functionality: While the ESP8266 and DHT11 combination is suitable for basic temperature and humidity monitoring, it may lack advanced features or capabilities required for more sophisticated applications, such as data logging, real-time analysis, or integration with other IoT devices.
9. Security Concerns: As with any IoT device, there are potential security risks associated with transmitting data over WiFi. Without proper security measures in place, there is a risk of unauthorized access or tampering with the system.
10. Difficulty in Calibration: Calibrating the DHT11 sensor for accurate readings can be challenging, especially in environments where precise measurements are required. Calibration may require additional equipment and expertise.

**CODE and Output:**

The Arduino Integrated Development Environment (IDE) is an open-source software platform used for writing, compiling, and uploading code to Arduino boards. It provides a user-friendly interface for writing and uploading code, making it accessible to beginners and experienced developers alike.

#include <ESP8266WiFi.h>

#include <ESP8266WiFiAP.h>

#include <ESP8266WiFiGeneric.h>

#include <ESP8266WiFiGratuitous.h>

#include <ESP8266WiFiMulti.h>

#include <ESP8266WiFiScan.h>

#include <ESP8266WiFiSTA.h>

#include <ESP8266WiFiType.h>

#include <WiFiClient.h>

#include <WiFiClientSecure.h>

#include <WiFiClientSecureBearSSL.h>

#include <WiFiServer.h>

#include <WiFiServerSecure.h>

#include <WiFiServerSecureBearSSL.h>

#include <WiFiUdp.h>

#include <DHT.h> // Including library for dht

#include <ESP8266WiFi.h>

String apiKey = "836FGI0478S0WEYT"; // Enter your Write API key from ThingSpeak

const char \*ssid = "Kasturi's Galaxy A32"; // replace with your wifi ssid and wpa2 key

const char \*pass = "vmny816045";

const char\* server = "api.thingspeak.com";

#define DHTPIN 0 //pin where the dht11 is connected

DHT dht(DHTPIN, DHT11);

WiFiClient client;

void setup()

{

Serial.begin(9600);

delay(10);

dht.begin();

Serial.println("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, pass);

while (WiFi.status() != WL\_CONNECTED)

{

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

}

void loop()

{

float h = dht.readHumidity();

float t = dht.readTemperature();

if (isnan(h) || isnan(t))

{

Serial.println("Failed to read from DHT sensor!");

return;

}

if (client.connect(server,80)) // "184.106.153.149" or api.thingspeak.com

{

String postStr = apiKey;

postStr +="&field1=";

postStr += String(t);

postStr +="&field2=";

postStr += String(h);

postStr += "\r\n\r\n";

client.print("POST /update HTTP/1.1\n");

client.print("Host: api.thingspeak.com\n");

client.print("Connection: close\n");

client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");

client.print("Content-Type: application/x-www-form-urlencoded\n");

client.print("Content-Length: ");

client.print(postStr.length());

client.print("\n\n");

client.print(postStr);

Serial.print("Temperature: ");

Serial.print(t);

Serial.print(" degrees Celcius, Humidity: ");

Serial.print(h);

Serial.println("%. Send to Thingspeak.");

}

client.stop();

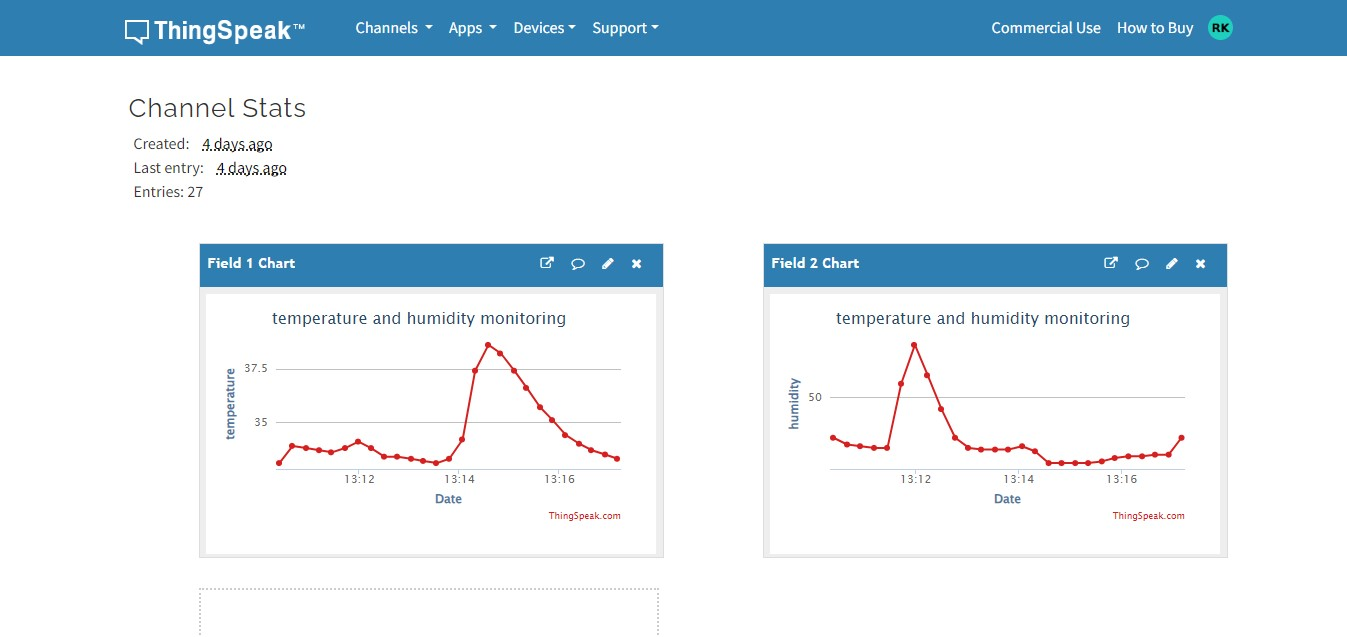
Serial.println("Waiting...");

// thingspeak needs minimum 15 sec delay between updates

delay(1000);

}



****

**Conclusion:**

The analysis conducted throughout this project report has highlighted the numerous advantages of Temperature and Humidity monitoring system using ESP8266 and DHT11. By harnessing the power of advanced technologies such as IoT, artificial intelligence, and data analytics can optimize their living spaces to suit their preferences and lifestyles while also contributing to sustainability efforts through energy conservation.