

Wi-Fi Weather Station using Arduino & ESP8266

Project Overview

This **Wi-Fi Weather Station** is designed to monitor and display real-time weather conditions, including **temperature, humidity, and atmospheric pressure**. The system uses an **ESP8266 Wi-Fi module** to send the sensor data to a web dashboard, allowing users to monitor weather conditions remotely.

Objectives

- ✓ **Measure real-time weather parameters** like temperature, humidity, and pressure.
- ✓ **Send data wirelessly** to a cloud service or a web dashboard.
- ✓ **Display weather data on an OLED/LCD screen** for local monitoring.
- ✓ **Provide historical weather data** using cloud storage or a local server.

Components Required

1. **Arduino Uno** – Main microcontroller for processing sensor data.
2. **ESP8266 (NodeMCU or ESP-01)** – Wi-Fi module for cloud connectivity.
3. **DHT11/DHT22 Sensor** – Measures temperature and humidity.
4. **BMP280/BME280 Sensor (Optional)** – Measures atmospheric pressure.
5. **OLED Display (0.96" I2C) / 16x2 LCD with I2C** – Displays local weather conditions.
6. **Jumper Wires & Breadboard** – For connections.
7. **Power Supply (5V/USB adapter)** – Powers the components.

Working Mechanism

1. **The sensors collect weather data** (temperature, humidity, and pressure).
2. **Arduino processes the data** and sends it to the ESP8266 module.
3. The **ESP8266 connects to Wi-Fi** and uploads the data to a cloud service (ThingSpeak, Firebase, or a web dashboard).
4. The **data is displayed on an OLED screen** for local monitoring.
5. The **user can view live weather data remotely** from a mobile or PC via the web interface.

Circuit Diagram

Connections for DHT11 Sensor:

DHT11 Arduino Uno

VCC 5V
GND GND
DATA Pin 2

Connections for ESP8266:

ESP8266 Arduino Uno

VCC 3.3V
GND GND
TX Pin 10 (Software Serial RX)
RX Pin 11 (Software Serial TX)

Connections for OLED Display (I2C):

OLED (I2C) Arduino Uno

VCC 5V
GND GND
SDA A4
SCL A5

Arduino Code for Wi-Fi Weather Station

This code reads data from the **DHT11 sensor** and uploads it to **ThingSpeak** using ESP8266.

```
#include <ESP8266WiFi.h>
#include <DHT.h>

#define DHTPIN 2          // Pin where DHT11 is connected
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

const char* ssid = "Your_WiFi_Name";          // Wi-Fi Name
const char* password = "Your_WiFi_Password"; // Wi-Fi Password
const char* server = "api.thingspeak.com";
String apiKey = "Your_ThingSpeak_API_Key";

WiFiClient client;
```

```
void setup() {
  Serial.begin(115200);
  dht.begin();

  // Connect to Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting...");
  }
  Serial.println("Connected to Wi-Fi!");
}

void loop() {
  float temp = dht.readTemperature();
  float hum = dht.readHumidity();

  if (isnan(temp) || isnan(hum)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  Serial.print("Temperature: ");
  Serial.print(temp);
  Serial.print(" °C, Humidity: ");
  Serial.print(hum);
  Serial.println(" %");

  if (client.connect(server, 80)) {
    String data = "GET /update?api_key=" + apiKey + "&field1=" +
String(temp) + "&field2=" + String(hum);
    client.print(data);
    client.stop();
  }

  delay(30000); // Upload data every 30 seconds
}
```

How to Use the System

1. **Upload the code to Arduino Uno.**
2. Connect your ESP8266 module to Wi-Fi.
3. **Go to ThingSpeak or Firebase** to view the real-time weather data.
4. The **OLED/LCD screen** will display local weather readings.
5. Open your **web dashboard** to monitor weather remotely.

Features & Benefits

- ✓ **Real-Time Monitoring** – Live weather data updates every 30 seconds.
 - ✓ **Wi-Fi Connectivity** – View data remotely on a web dashboard.
 - ✓ **Scalability** – Can be integrated with IoT cloud platforms.
 - ✓ **Energy Efficient** – Uses low-power ESP8266 for connectivity.
-

Future Enhancements

Add **Rain & Wind Sensors** for more detailed weather data.
Send **Email/SMS Alerts** for extreme weather conditions.
Display a **Weather Forecast** using OpenWeather API.
Control **Home Appliances (Fans, AC)** based on temperature.



BINARY BRAINS