

# ESP32 Mini Weather Station with DHT11 & OLED Display

---

## 1. Objective

Create a simple weather station using the ESP32 microcontroller that:

- Reads temperature and humidity from a DHT11 sensor.
- Displays the values on a 0.96-inch OLED screen.
- Also outputs the readings to the Serial Monitor.

## 2. Required Hardware Components

Component	Description
ESP32	Microcontroller with Wi-Fi & GPIOs
DHT11	Digital temperature & humidity sensor
SSD1306 OLED	128x64 I2C-based OLED display
Jumper wires	For making connections
Breadboard	(Optional) For prototyping

## 3. Wiring Diagram & Pin Configuration

DHT11 Sensor (3-pin version)

- VCC → 3.3V or 5V (ESP32 supports both, check your sensor label)
- GND → GND
- DATA → GPIO 4

Note: If your DHT11 has a 4th pin (NC), it's not connected internally. Ignore it.

0.96" OLED Display (SSD1306, I<sup>2</sup>C)

- VCC → 3.3V (or 5V if your module supports it)
- GND → GND
- SDA → GPIO 21 (ESP32 default I2C SDA)
- SCL → GPIO 22 (ESP32 default I2C SCL)

The I2C address for most SSD1306 OLEDs is 0x3C.

## 4. Library Explanation

Library	Purpose
Adafruit_Sensor.h	Base class for sensor data
DHT.h & DHT_U.h	For DHT11 sensor operations
Wire.h	Manages I2C communication
Adafruit_GFX.h	Graphics support (fonts, text rendering, etc.)
Adafruit_SSD1306.h	Driver for SSD1306 OLED module

## 5. Full Code Explanation – Line by Line

```
#include <Arduino.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

// --- Pin Definitions ---
#define DHTPIN 4           // GPIO for DHT11 data
#define DHTTYPE DHT11

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1     // Not used with I2C

// --- Initialize Objects ---
DHT dht(DHTPIN, DHTTYPE);
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

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

  // Initialize OLED
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3C for 128x64
    Serial.println(F("SSD1306 allocation failed"));
    for (;;);
  }
}
```

```

display.clearDisplay();
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.setCursor(0, 0);
display.println("Mini Weather Station");
display.display();
delay(2000);
}

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

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

    // Print to Serial Monitor
    Serial.print("Temp: ");
    Serial.print(temp);
    Serial.print(" °C | Hum: ");
    Serial.print(hum);
    Serial.println(" %");

    // Show on OLED
    display.clearDisplay();
    display.setCursor(0, 0);
    display.setTextSize(1);
    display.println("Mini Weather Station");

    display.setCursor(0, 20);
    display.setTextSize(2);
    display.print("T:");
    display.print(temp);
    display.print("C");

    display.setCursor(0, 45);
    display.print("H:");
    display.print(hum);
    display.print("%");

    display.display();
    delay(2000); // Read every 2 seconds
}

```

## Include Libraries

```
#include <Arduino.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
```

- These import external library functions for sensor and display handling.
- **Wire.h** is the I2C protocol handler used by OLED.

## Define Constants

```
// --- Pin Definitions ---
#define DHTPIN 4           // GPIO for DHT11 data
#define DHTTYPE DHT11

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1     // Not used with I2C
```

## Create Object Instances

```
DHT dht(DHTPIN, DHTTYPE);
// Create DHT object with defined pin and type
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET); // OLED
object
```

- **dht** is now ready to talk to the sensor.
- **display** is initialized with screen size and I2C interface (**Wire**).

## 6. setup() – Initializing Hardware

```
Serial.begin(115200);
dht.begin();
```

- Serial is useful for debugging and live data viewing.
- **dht.begin()** starts internal timing and calibration.

```
// Initialize OLED
if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3C for 128x64
  Serial.println(F("SSD1306 allocation failed"));
  for (;;);
}
```

- Checks if OLED is connected at I2C address `0x3C`.
- If not found, it prints an error and stops execution.

```
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.setCursor(0, 0);
display.println("Mini Weather Station");
display.display();
delay(2000);
```

## 7. loop() – Main Repeating Function

### Reading Data

```
float temp = dht.readTemperature(); // Celsius
float hum = dht.readHumidity();
```

- These functions may return NaN (Not a Number) if the reading fails.

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

- Prevents writing invalid data to screen/Serial.

### Serial Output

```
// Print to Serial Monitor
Serial.print("Temp: ");
Serial.print(temp);
Serial.print(" °C | Hum: ");
Serial.print(hum);
Serial.println(" %");
```

- Helps monitor data via Serial Monitor in real time.

## OLED Display Update

## 8. Visual Output (OLED Screen)

```
| Mini Weather Station |
|                       |
| T: 26.0C             |
| H: 61.0%             |
```

## 9. Technical Insights

- DHT11 Update Rate: About 1 reading per second max. Delay is 2 seconds to avoid read errors.
- OLED Buffering: `Adafruit_SSD1306` uses a framebuffer, so nothing appears until `display.display()` is called.
- Text Scaling: `setTextSize()` scales the font size in both x and y.

## 10. Loop Cycle

Every 2 seconds:

1. Clear the screen
2. Read temperature and humidity
3. Validate readings
4. Print to Serial
5. Display values on OLED

Temp: 31.60 °C | Hum: 69.00 %

Temp: 31.80 °C | Hum: 69.00 %

Temp: 31.80 °C | Hum: 69.00 %

