

ESP32 Mini Weather Station with OLED Display and SD Card Logging

1. Overview

This project reads **temperature and humidity** data from a **DHT11 sensor**, displays it on a **0.96" I2C OLED screen**, and logs the data with a timestamp to an **SD card** in CSV format for future analysis. It updates every **5 seconds**.

2. Hardware Components and Pin Connections

Component	ESP32 GPIO Pin	Notes
DHT11	GPIO 4	Data pin connected to GPIO 4
OLED (SSD1306)	I2C - SDA: GPIO 21 I2C - SCL: GPIO 22	Default I2C on ESP32
SD Card Module	CS: GPIO 5 SCK: GPIO 18 MOSI: GPIO 23 MISO: GPIO 19	SPI pins for SD card

Use **level shifter or voltage divider** for SD card if using 5V modules on 3.3V ESP32.

3. Line-by-Line Code Explanation

```
#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>
#include <SPI.h>
#include <SD.h>

// --- Pin Definitions ---
#define DHTPIN 4
#define DHTTYPE DHT11
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
#define SD_CS 5 // SD Card Chip Select
```

```

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

unsigned long startTime;

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

    // Initialize OLED
    if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
        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);

    // Initialize SD Card
    if (!SD.begin(SD_CS)) {
        Serial.println("SD Card initialization failed!");
        display.setCursor(0, 20);
        display.println("SD Init Failed!");
        display.display();
        while (true); // Halt
    }
    Serial.println("SD Card ready.");

    // Create or open file
    if (!SD.exists("/weather_log.csv")) {
        logFile = SD.open("/weather_log.csv", FILE_WRITE);
        logFile.println("Time(s),Temperature(C),Humidity(%)");
        logFile.close();
    }
}

```

```

    startTime = millis();
}

void loop() {
    float temp = dht.readTemperature();
    float hum = dht.readHumidity();
    unsigned long currentTime = (millis() - startTime) / 1000;

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

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

    // OLED Output
    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();

    // SD Card Logging
    String dataString = String(currentTime) + "," + String(temp) + "," +
String(hum);
    logFile = SD.open("/weather_log.csv", FILE_APPEND);

```

```

if (logFile) {
    logFile.println(dataString);
    logFile.close();
} else {
    Serial.println("Failed to open log file!");
}

delay(5000); // Log every 5 seconds
}

```

EXPLANATION

```

#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>
#include <SPI.h>
#include <SD.h>

```

- These lines include all required libraries:
 - **DHT** and **DHT_U**: For DHT11 sensor.
 - **Wire**: I2C communication.
 - **Adafruit_GFX** and **Adafruit_SSD1306**: For OLED display.
 - **SPI** and **SD**: For SD card communication and file handling.

Pin Definitions

```

// --- Pin Definitions ---
#define DHTPIN 4
#define DHTTYPE DHT11
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
#define SD_CS 5 // SD Card Chip Select

```

- **DHTPIN (GPIO 4)**: Connected to the DHT11's data pin.
- **OLED_RESET**: Set to -1 (not connected; reset handled by I2C).
- **SD_CS (GPIO 5)**: Chip Select pin for SD card SPI communication.

Object Initialization

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

- **dht**: An instance of the DHT sensor.
- **display**: OLED screen object using I2C.
- **logFile**: File object for reading/writing to SD card.
- **startTime**: Stores millis() at startup for time tracking.

```
void setup()
```

– Initialization

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

- Begins serial communication.
- Initializes the DHT sensor.

OLED Initialization

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

- Initializes the OLED at I2C address **0x3C**.
- If initialization fails, prints an error and halts the system.

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

- Clears OLED, sets text properties, prints a welcome message, and displays it for 2 seconds.

SD Card Initialization

```
// Initialize SD Card
if (!SD.begin(SD_CS)) {
  Serial.println("SD Card initialization failed!");
  display.setCursor(0, 20);
  display.println("SD Init Failed!");
  display.display();
  while (true); // Halt
}
```

- Initializes the SD card using chip select pin (GPIO 5).
- If SD card fails, displays error and stops further execution.

```
Serial.println("SD Card ready.");

// Create or open file
if (!SD.exists("/weather_log.csv")) {
  logFile = SD.open("/weather_log.csv", FILE_WRITE);
  logFile.println("Time(s),Temperature(C),Humidity(%)");
  logFile.close();
}
```

- If the log file doesn't exist, it creates one and adds a CSV header.

```
startTime = millis();
```

- Records the current time in milliseconds at the start to compute elapsed time in seconds.

```
void loop()
```

– Repeating Logic

```
float temp = dht.readTemperature();
float hum = dht.readHumidity();
unsigned long currentTime = (millis() - startTime) / 1000;
```

- Reads temperature and humidity values.
- Calculates **time in seconds** since startup.

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

- If the DHT sensor gives invalid reading (**NaN**), the loop skips further execution.

Serial Monitor Output

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

- Displays the sensor values in a human-readable format in Serial Monitor.

OLED Display Output

```
// OLED Output
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();
```

- Clears OLED and prints updated sensor readings.
- Temperature and humidity are displayed in larger font.

SD Card Logging

```
// SD Card Logging
String dataString = String(currentTime) + "," + String(temp) + "," +
    String(hum);
logFile = SD.open("/weather_log.csv", FILE_APPEND);
if (logFile) {
    logFile.println(dataString);
    logFile.close();
} else {
    Serial.println("Failed to open log file!");
}
```

- Formats sensor values and timestamp as a CSV string.
- Opens the file in append mode and adds a new line for every reading.
- Closes the file to avoid corruption.

```
delay(5000); // Log every 5 seconds
```

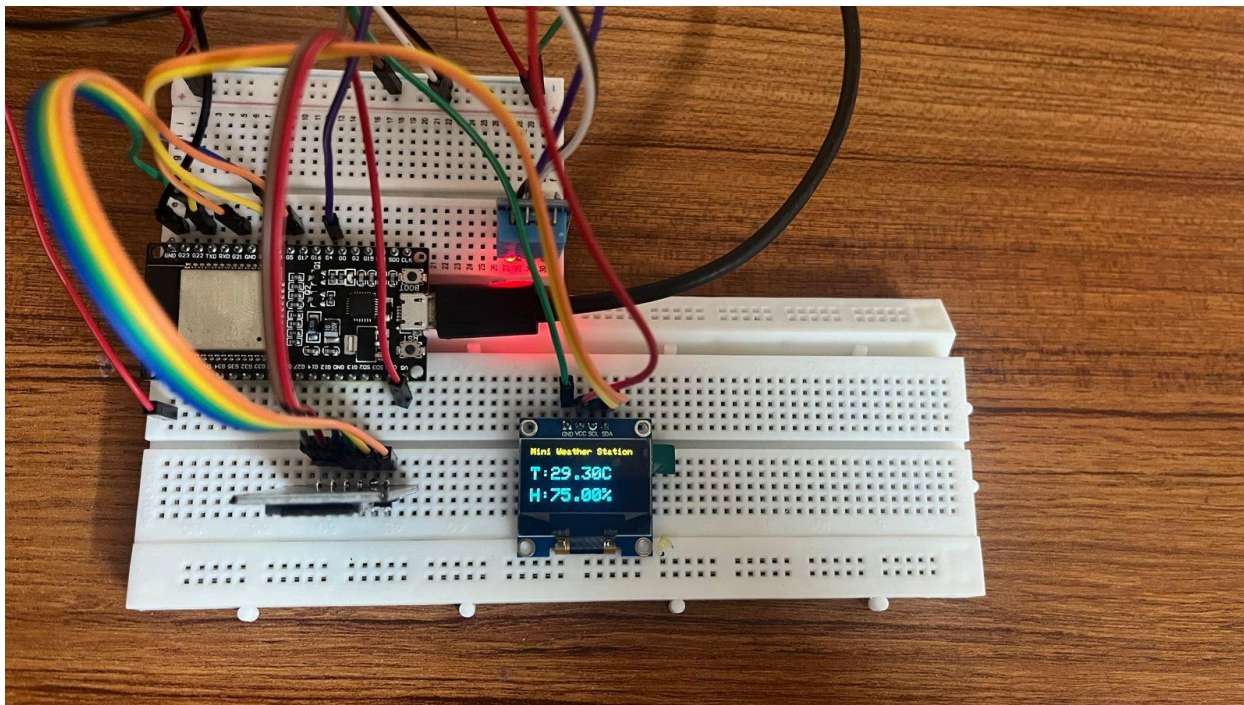
- Waits 5 seconds before the next reading and log entry.

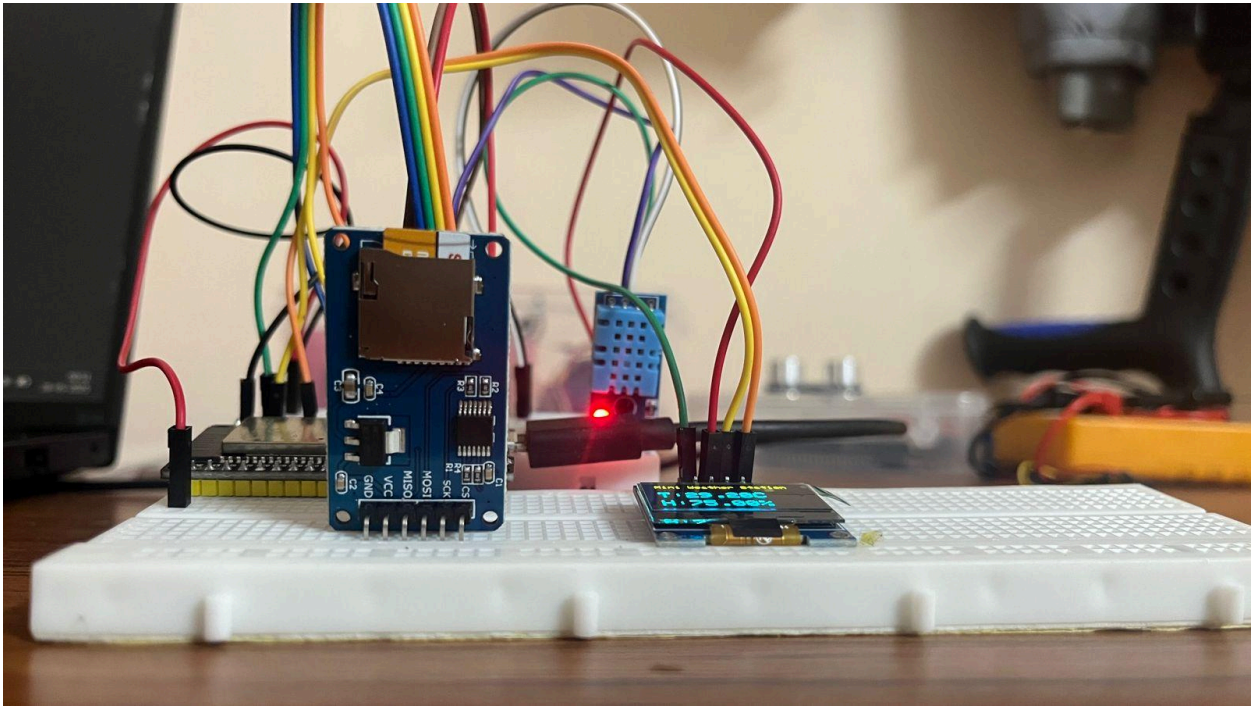
4. Sample Output in CSV File

```
Time(s),Temperature(C),Humidity(%)
5,26.00,55.00
10,26.10,56.00
15,26.20,55.50
```

5. Summary of Operation

Step	Operation
1	ESP32 powers up, initializes Serial, DHT11, OLED, and SD card
2	Displays "Mini Weather Station" on OLED
3	Reads temperature and humidity from DHT11
4	Displays current values on OLED and prints to Serial Monitor
5	Saves each data point with a timestamp to SD card every 5 seconds





weather... Saved to this PC

File Home Insert Page Layout Formulas Data Review View Help

Clipboard Font Alignment Number Styles Cells Editing Add-ins

H13

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Time(s)	Temperature(°C)	Humidity(%)																
2	0	0.2	0																
3	5	29.8	75																
4	10	29.7	75																
5	15	29.1	75																
6	20	29.3	75																
7	25	29.2	75																
8	30	29.8	74																
9	35	29.4	75																
10	40	29.7	75																
11	45	29	75																
12	0	29.2	75																
13	5	29.4	75																
14	10	29.5	75																
15	15	29.7	74																
16	20	29	74																
17	0	0.2	0																
18	5	29.1	75																
19	10	29.1	75																
20	15	29	75																
21	20	29.5	75																
22	25	29.8	75																
23	30	29.4	75																
24	35	29.3	75																

weather_log

Ready Accessibility: Unavailable

ENG IN 00:52 28-05-2025