



# VISVESVARAYA NATIONAL INSTITUTE OF TECHNOLOGY, NAGPUR

---

## Embedded Systems (ECP403) Lab Report

---

Submitted By :  
Shruti Murarka (BT18ECE099)  
Semester 5  
Electronics and Communication Engineering Dept.

Submitted To :  
Dr. Ankit A Bhurane  
Course Instructor

## ESP32 based Web Server for Temperature and Humidity Monitor using DHT11 Sensor

**Aim:** To monitor real time temperature & humidity values on a webserver using DHT11 sensor & ESP32.

**DHT11 Sensor Specifications:** The DHT11 is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin. The only real downside of this sensor is you can only get new data from it once every 2 seconds, so when using our library, sensor readings can be up to 2 seconds old. Compared to the DHT22, this sensor is less precise, less accurate and works in a smaller range of temperature/humidity, but its smaller and less expensive.

DHT11 Specifications:

- Low cost
- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 20-80% humidity readings with 5% accuracy
- Good for 0-50°C temperature readings  $\pm 2^{\circ}\text{C}$  accuracy
- No more than 1 Hz sampling rate (once every second)
- Body size 15.5mm x 12mm x 5.5mm
- 4 pins with 0.1" spacing

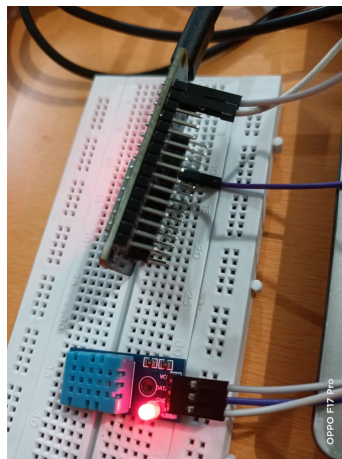


Figure 1: Connection between ESP32 & DHT11

**Connections:** Install "DHT Sensor" Library for DHT Sensor.

Arduino IDE > Sketch > Include Library > Manage Libraries > DHT SENSOR.

DHT11 sensor normally comes with only three pins: VCC, GND and the Data line. Data Line is connected to GPIO4 (D4).

The pins are labeled so that you know how to wire them. Additionally, many of these modules already come with an internal pull up resistor, so you don't need to add one to the circuit.

**Code :**

---

```
1 #include "WiFi.h"
2 #include "DHT.h"
3
4 #define DHTPIN 4      // Digital pin connected to the DHT sensor
5 #define DHTTYPE DHT11 // DHT 11
6
7 const char* ssid = "Shruti_ESP";
8 const char* pass = "Shubh@123";
9
10 WiFiServer server(80);
11
12 DHT dht(DHTPIN, DHTTYPE);
13 String output = "";
14 String output1 = "";
15 String html = "<!DOCTYPE html> \
16 <html> \
17 <head> <meta http-equiv=\"refresh\" content=\"1\"> \
18 </head> \
19 <body> \
20 Temperature(in Celsius): ";
21
22 String html2 = "</body> \
23 </html> ";
24
25 void setup() {
26   Serial.begin(115200);
27   WiFi.softAP(ssid, pass);
28   IPAddress IP = WiFi.softAPIP();
29   Serial.print("AP IP address: ");
30   Serial.println(IP);
31   server.begin();
32   dht.begin();
33 }
34
```

```

35 void loop() {
36
37     float h = dht.readHumidity();    //Read Humidity
38     float t = dht.readTemperature(); // Read temperature as Celsius
39     float f = dht.readTemperature(true); // Read temperature as ...
        Fahrenheit (isFahrenheit = true)
40
41     WiFiClient client = server.available();
42     if (isnan(h) || isnan(t) || isnan(f)) {
43         Serial.println(F("Failed to read from DHT sensor!"));
44         return;}
45     float hif = dht.computeHeatIndex(f, h); // Compute heat ...
        index in Fahrenheit
46     float hic = dht.computeHeatIndex(t, h, false); // Compute ...
        heat index in Celsius (isFahreheit = false)
47     Serial.print(F("Humidity: "));
48     Serial.print(h);
49     Serial.print(F("% Temperature: "));
50     Serial.print(t);
51     Serial.print(F("C "));
52     Serial.print(f);
53     Serial.print(F("F Heat index: "));
54     Serial.print(hic);
55     Serial.print(F("C "));
56     Serial.print(hif);
57     Serial.println(F("F"));
58     output = String(t) + "<br>";
59     output1 = String(h) + "<br>";
60     client.print(html + output + html2);
61     client.print("Humidity(in %): " + output1 + html2);
62     delay(1500);
63 }

```

## Adding Style to Webpage

```

1 #include "WiFi.h"
2 #include "DHT.h"
3
4 #define DHTPIN 4 // Digital pin connected to the DHT sensor
5 #define DHTTYPE DHT11 // DHT 11
6
7 const char* ssid = "Shruti_ESP";
8 const char* pass = "Shubh@123";
9
10 WiFiServer server(80);
11 String output = "";
12 String output1 = "";
13

```

```
14 DHT dht (DHTPIN, DHTTYPE);
15
16 void setup() {
17     Serial.begin(115200);
18     WiFi.softAP(ssid, pass);
19     IPAddress IP = WiFi.softAPIP();
20     Serial.print("AP IP address: ");
21     Serial.println(IP);
22     server.begin();
23     dht.begin();
24 }
25
26 void loop() {
27
28     float h = dht.readHumidity();    //Read Humidity
29     float t = dht.readTemperature(); // Read temperature as Celsius
30     float f = dht.readTemperature(true); // Read temperature as ...
        Fahrenheit (isFahrenheit = true)
31
32     if (isnan(h) || isnan(t) || isnan(f)) {
33         Serial.println(F("Failed to read from DHT sensor!"));
34         return;
35     }
36     float hif = dht.computeHeatIndex(f, h); // Compute heat ...
        index in Fahrenheit
37     float hic = dht.computeHeatIndex(t, h, false); // Compute ...
        heat index in Celsius (isFahreheit = false)
38     Serial.print(F("Humidity: "));
39     Serial.print(h);
40     Serial.print(F("% Temperature: "));
41     Serial.print(t);
42     Serial.print(F("C "));
43     Serial.print(f);
44     Serial.print(F("F Heat index: "));
45     Serial.print(hic);
46     Serial.print(F("C "));
47     Serial.print(hif);
48     Serial.println(F("F"));
49     output = String(t) + "<br>";
50     output1 = String(h) + "<br>";
51     WiFiClient client = server.available();
52     if(client)
53     {
54         Serial.println("Web Client connected ");
55         String request = client.readStringUntil('\r');
56         client.println("HTTP/1.1 200 OK");
57         client.println("Content-type:text/html");
58         client.println("Connection: close");
59         client.println();
60         client.println("<!DOCTYPE html><html>");
```

```

60     client.println("<head><meta name=\"viewport\" ...
        content=\"width=device-width, initial-scale=1\">");
61     client.println("<link rel=\"icon\" href=\"data:,\">");
62
63     //Adding styles to a web page to look more professional
64     client.println("<style>body { text-align: center; ...
        font-family: \"Arial\", Arial;}");
65     client.println("table { border-collapse: collapse; ...
        width:40%; margin-left:auto; ...
        margin-right:auto;border-spacing: 2px;background-color: ...
        white;border: 4px solid green; }");
66     client.println("th { padding: 20px; background-color: ...
        #008000; color: white; }");
67     client.println("tr { border: 5px solid green; padding: 2px; }");
68     client.println("tr:hover { background-color:yellow; }");
69     client.println("td { border:4px; padding: 12px; }");
70     client.println(".sensor { color:white; font-weight: bold; ...
        background-color: #bcbcbc; padding: 1px; }");
71
72     client.println("</style></head><body><h1>ESP32 Web Server ...
        </h1>");
73     client.println("<h2>DHT11/DHT22</h2>");
74     client.println("<table><tr><th>MEASUREMENT</th><th>VALUE</th></tr>");
75     client.println("<tr><td>Temp. Celsius</td><td><span ...
        class=\"sensor\">");
76     client.println(dht.readTemperature());
77     client.println(" *C</span></td></tr>");
78     client.println("<tr><td>Temp. Fahrenheit</td><td><span ...
        class=\"sensor\">");
79     client.println(1.8 * dht.readTemperature() + 32);
80     client.println(" *F</span></td></tr>");
81     client.println("<tr><td>Humidity</td><td><span ...
        class=\"sensor\">");
82     client.println(dht.readHumidity());
83     client.println(" %</span></td></tr>");
84     client.println("</body></html>");
85     client.stop();
86     client.println();
87     Serial.println("Client disconnected.");
88     Serial.println("");
89 }
90 delay(1500);
91 }

```

**Output:** Arduino IDE is used to compile code.

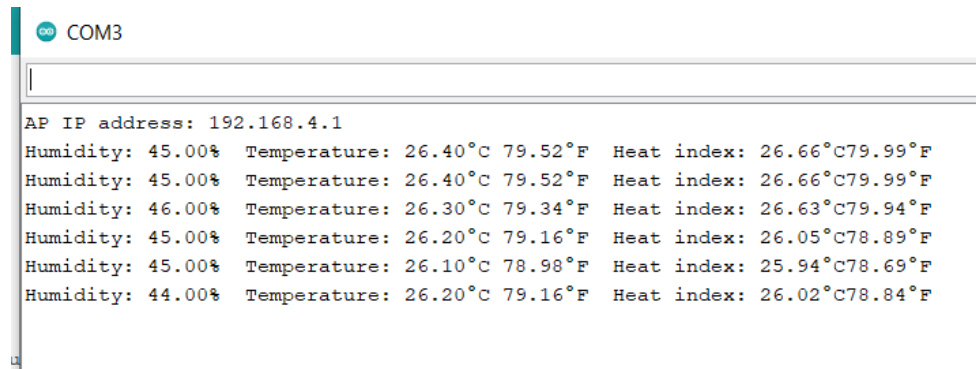
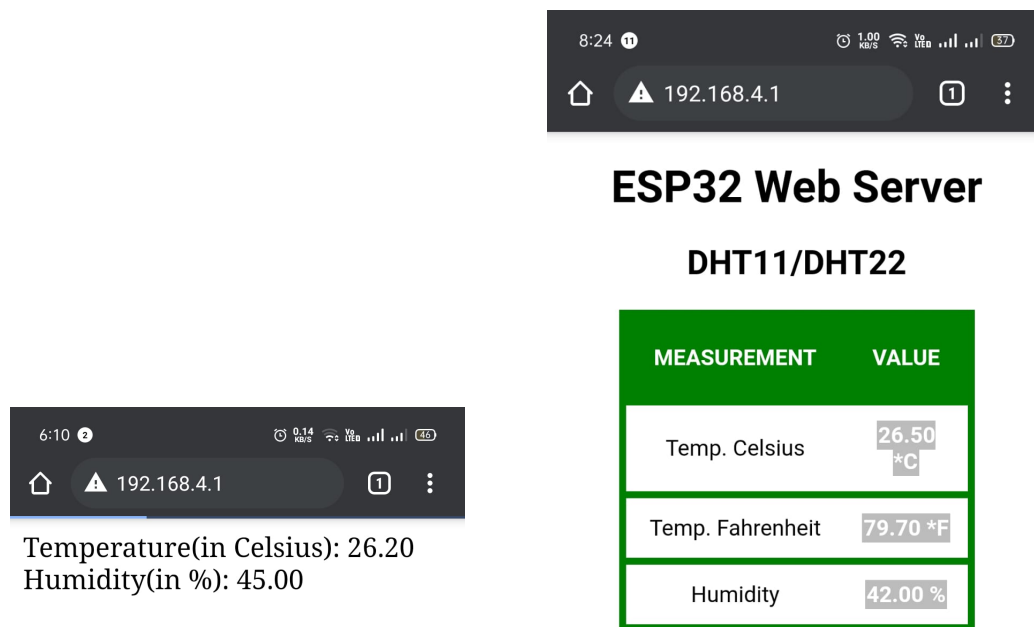


Figure 2: Serial Monitor



(a) Simple WebServer.

(b) Advance WebServer

Figure 3: DHT11 with ESP32 on WebServer.

**Discussions & Observations:** This project uses ESP32 as the control device & creates a Web Server. When any connected device accesses this web server, ESP32 reads in temperature & relative humidity from the DHT11 sensor & sends it to the web browser of that device.

**Conclusion:** Hardware Demonstration: [\[Video\]](#).