Real Time Temperature and Humidity Reading with Arduino and DHT11 sensor

By Dhruv Kushalkar

Introduction

In this project I have used NodeMCU which runs on ESP8266 WiFi shield and accepts Arduino codes. I have used DHT11 sensor which senses real time temperature and humidity.

Arduino Code

```
#include <Arduino.h>

#include <ESP8266WiFi.h>

#include <Hash.h>

#include <ESPAsyncTCP.h>

#include <ESPAsyncWebServer.h>

#include <Adafruit_Sensor.h>

#include <DHT.h>


// Replace with your network credentials

const char* ssid = "Room 1";

const char* password = "aryan@9545529735";


#define DHTPIN 5     // Digital pin D1 connected to the DHT sensor


// Uncomment the type of sensor in use:

#define DHTTYPE    DHT11     // DHT 11

//#define DHTTYPE    DHT22     // DHT 22 (AM2302)

//#define DHTTYPE    DHT21     // DHT 21 (AM2301)


DHT dht(DHTPIN, DHTTYPE);


// current temperature & humidity, updated in loop()

float t = 0.0;
```

```
float h = 0.0;

// Create AsyncWebServer object on port 80
AsyncWebServer server(80);

// Generally, you should use "unsigned long" for variables that hold time
// The value will quickly become too large for an int to store
unsigned long previousMillis = 0;    // will store last time DHT was updated

// Updates DHT readings every 10 seconds
const long interval = 10000;

const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE HTML><html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <style>
    html {
     font-family: Arial;
     display: inline-block;
     margin: 0px auto;
     text-align: center;
    }
    h2 { font-size: 3.0rem; }
    p { font-size: 3.0rem; }
    .units { font-size: 1.2rem; }
    .dht-labels{
      font-size: 1.5rem;
      vertical-align:middle;
      padding-bottom: 15px;
    }
  </style>
</head>
<body>
```

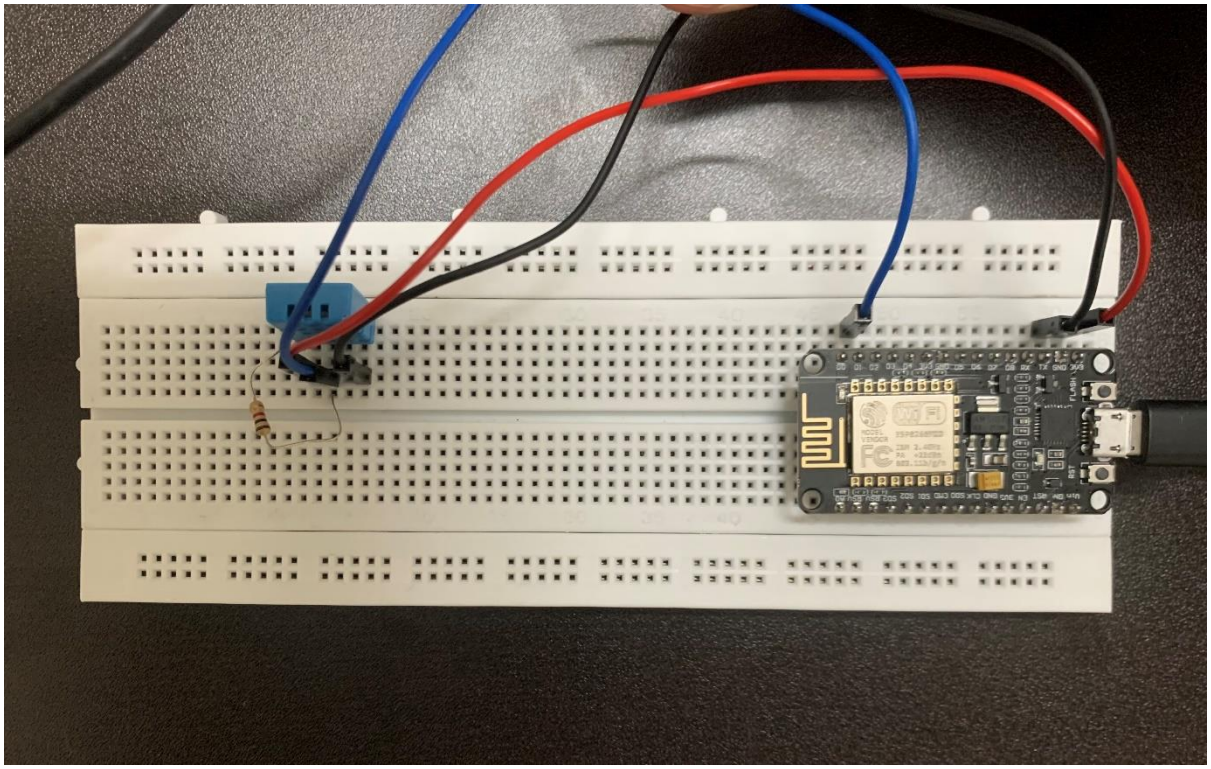
```
  <h2>ESP8266 DHT Server</h2>
  <p>
   <i class="fas fa-thermometer-half" style="color:#059e8a;"></i>
   <span class="dht-labels">Temperature</span>
   <span id="temperature">%TEMPERATURE%</span>
   <sup class="units">°C</sup>
  </p>
  <p>
   <i class="fas fa-tint" style="color:#00add6;"></i>
   <span class="dht-labels">Humidity</span>
   <span id="humidity">%HUMIDITY%</span>
   <sup class="units">%</sup>
  </p>

</body>
<script>
setInterval(function ( ) {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("temperature").innerHTML = this.responseText;
    }
  };
  xhttp.open("GET", "/temperature", true);
  xhttp.send();
}, 10000 ) ;

setInterval(function ( ) {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("humidity").innerHTML = this.responseText;
    }
  };
  xhttp.open("GET", "/humidity", true);
```

```
  xhttp.send();
}, 10000 ) ;
</script>
</html>)rawliteral";

// Replaces placeholder with DHT values
String processor(const String& var){
  //Serial.println(var);
  if(var == "TEMPERATURE"){
    return String(t);
  }
  else if(var == "HUMIDITY"){
    return String(h);
  }
  return String();
}

void setup(){
  // Serial port for debugging purposes
  Serial.begin(115200);
  dht.begin();

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

  // Print ESP8266 Local IP Address
  Serial.println(WiFi.localIP());

  // Route for root / web page
  server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
```

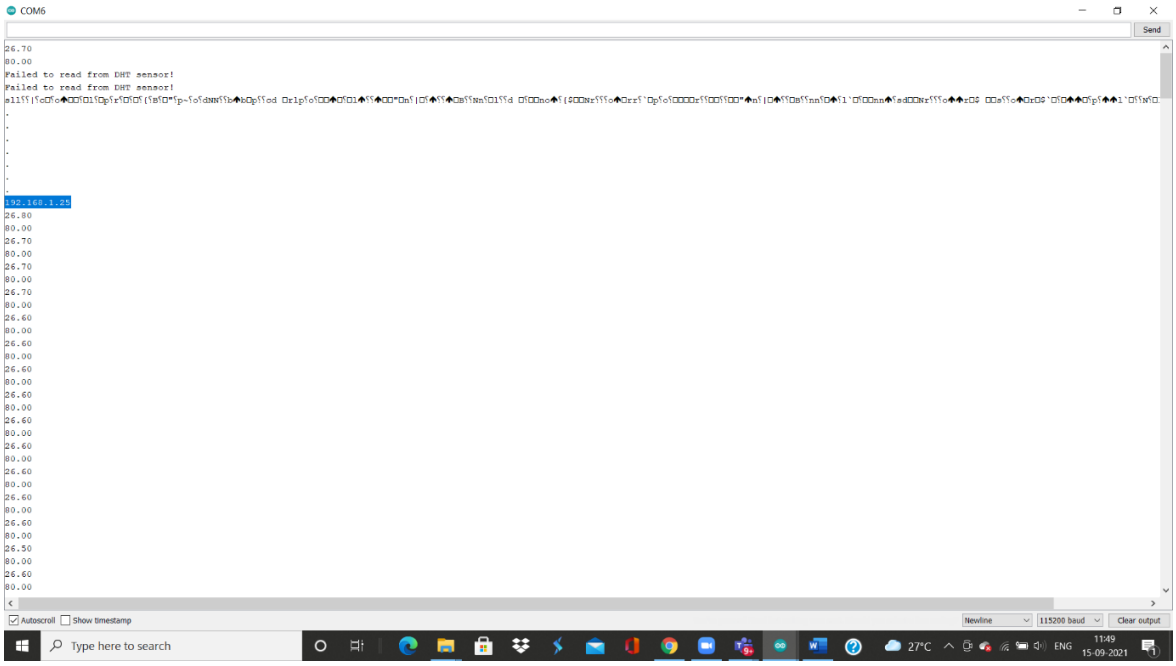
```
  request->send_P(200, "text/html", index_html, processor);
 });
 server.on("/temperature", HTTP_GET, [](AsyncWebServerRequest *request){
  request->send_P(200, "text/plain", String(t).c_str());
 });
 server.on("/humidity", HTTP_GET, [](AsyncWebServerRequest *request){
  request->send_P(200, "text/plain", String(h).c_str());
 });

 // Start server
 server.begin();
}

void loop(){
 unsigned long currentMillis = millis();
 if (currentMillis - previousMillis >= interval) {
  // save the last time you updated the DHT values
  previousMillis = currentMillis;
  // Read temperature as Celsius (the default)
  float newT = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  //float newT = dht.readTemperature(true);
  // if temperature read failed, don't change t value
  if (isnan(newT)) {
   Serial.println("Failed to read from DHT sensor!");
  }
  else {
   t = newT;
   Serial.println(t);
  }
  // Read Humidity
  float newH = dht.readHumidity();
  // if humidity read failed, don't change h value
  if (isnan(newH)) {
   Serial.println("Failed to read from DHT sensor!");
```

```
    }
    else {
      h = newH;
      Serial.println(h);
    }
  }
}
```

Circuit screenshot



Output on serial monitor



Need of Wifi Sheild

Here I have connected my circuit to my common WiFi. The Arduino(NodeMCU) sends an **IP address** which we can type on the web server to view the desired output. Thus a wifi shield is necessary in my project.

Output on web (IP address)