

SERVIDOR WEB PARA MONITOREAR TEMPERATURA Y HUMEDAD USANDO AJAX MEDIANTE EL SENSOR DHT11

SENSORES DHT11, DHT22

Los sensores DHT11 y DHT22 son sensores digitales de temperatura y humedad, fáciles de implementar con cualquier microcontrolador. Utiliza un sensor capacitivo de humedad y un termistor para medir el aire circundante y solo un pin para la lectura de los datos. Tal vez la desventaja de estos es la velocidad de las lecturas y el tiempo que hay que esperar para tomar nuevas lecturas (nueva lectura después de 2 segundos), pero esto no es tan importante puesto que la Temperatura y Humedad son variables que no cambian muy rápido en el tiempo. Para evitar que se haga lento el servidor web, realizaremos en este ejemplo las mediciones cada 100 ms.

El sensor DHT11 trabaja con un rango de medición de temperatura de 0 a 50 °C con precisión de ± 2.0 °C y un rango de humedad de 20% a 90% RH con precisión de 4% RH. Los ciclos de lectura debe ser como mínimo 1 o 2 segundos.

El sensor DHT22 tiene un rango de medición de temperatura es de -40°C a 80 °C con precisión de ± 0.5 °C y rango de humedad de 0 a 100% RH con precisión de 2% RH, el tiempo entre lecturas debe ser de 2 segundos.

LISTA DE MATERIALES:

- 1 Wemos D1 R1
- 1 protoboard
- 1 sensor DHT11
- 1 cables USB a USB micro
- 1 resistor de 4.7 KOhms
- 3 alambres dupont hembra-hembra que trae el módulo sensor
- 2 alambres dupont macho-macho color NARANJA
- 2 alambres dupont macho-macho color ROJO
- 1 alambre dupont macho-macho color NEGRO

INDICACIONES: Teclear el siguiente código en Platformio.

```
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <Hash.h>
#include <ESPAsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
```

```
const char* ssid = "MEGACABLE-F82T";
const char* password = "x423YrSu";
#define DHTPIN 4 // pin digital conectado al pin de DATA del sensor DTH 11
// Marcar con comentarios el sensor que no se esté utilizando:
```

```

#define TIPO_DHT DHT11 // DHT 11
// #define TIPO_DHT DHT22 // DHT 22 (AM2302)
// #define TIPO_DHT DHT21 // DHT 21 (AM2301)
DHT dht(DHTPIN, TIPO_DHT);
// temperatura y humedad, actualizados en el lazo
float t = 0.0;
float h = 0.0;
// Crear el objeto AsyncWebServer en el puerto 80
AsyncWebServer server(80);
// El tiempo por lo general se debería declarar "unsigned long" para variables que implican tiempo
// El valor llegará a ser muy grande al transcurrir el tiempo si declaramos int puede no alcanzar
unsigned long previosMilisegundos = 0; // almacenará el último tiempo en que DHT fue actualizado
// Actualiza DTH cada 1 segundo
const long interval = 1000; // intervalo para el monitor de arduino
const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE HTML><html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.7.2/css/all.css"
integrity="sha384-
fnmOCqbTIWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr"
crossorigin="anonymous">
<style>
html {
font-family: Arial;
display: inline-block;
margin: 0px auto;
text-align: center;
}
h2 { font-size: 3.0rem; }

p { font-size: 3.0rem; }
.unidades { font-size: 1.2rem; }
.etiquetas-DTH{
font-size: 1.5rem;
vertical-align: middle;
padding-bottom: 15px;
}
</style>
</head>
<body>
<h2>Servidor web con sensor DHT </h2>
<p>
<!-- para escoger el color del ícono de temperatura-->
<i class="fas fa-thermometer-half" style="color:#059e8a;"></i>
<!-- span aplica estilo al texto-->
<span class="etiquetas-DTH">Temperatura</span>
<span id=temperatura>%TEMPERATURA%</span>

```

```

<!-- unidad de grados como superíndice-->
<sup class="unidades">&deg;C</sup>
</p>
<p>
<!-- para escoger el color del ícono de humedad -->
<i class="fas fa-tint" style="color:#00add6;"></i>
<span class="etiquetas-DTH">Humedad</span>
<span id="humedad">%HUMEDAD%</span>
<sup class="unidades">%</sup>
</p>
</body>
<script>
setInterval(function ( ) {
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
if (this.readyState == 4 && this.status == 200) {
document.getElementById("temperatura").innerHTML = this.responseText;
}
};
xhttp.open("GET", "/temperatura", true);
xhttp.send();
}, 100 ) ; /* se efectúa la lectura de temp cada 100 ms */

setInterval(function ( ) {
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
if (this.readyState == 4 && this.status == 200) {
document.getElementById("humedad").innerHTML = this.responseText;
}
};
xhttp.open("GET", "/humedad", true);
xhttp.send();
}, 100 ) ; /* se efectúa la lectura de humedad cada 100 ms */
</script>
</html>rawliteral";
// La función propia processor reemplaza marcadores de posición
// TEMPERATURA y HUMEDAD con los valores de DTH
String processor(const String& var){

if(var == "TEMPERATURA"){
return String(t);
}
else if(var == "HUMEDAD"){
return String(h);
}
return String();
}
// -----
void setup(){

```

```

// Puerto serie para propósitos de grabación depuración
// y comprobación
Serial.begin(115200);
dht.begin();
// Conectar a Wi-Fi
WiFi.begin(ssid, password);
Serial.println("Conectando a la red WiFi");
while (WiFi.status() != WL_CONNECTED) {
  delay(1000);
  Serial.print(".");
}
// Presenta la IP local de ESP8266
Serial.println(WiFi.localIP());
// Ruta para la raíz / de la página web
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
  request->send_P(200, "text/html", index_html, processor);
}); // este 200 es de OK
server.on("/temperatura", HTTP_GET, [](AsyncWebServerRequest *request){
  request->send_P(200, "text/plain", String(t).c_str());
}); // este 200 es de OK
server.on("/humedad", HTTP_GET, [](AsyncWebServerRequest *request){
  request->send_P(200, "text/plain", String(h).c_str());
}); // este 200 es de OK
// Inicia el servidor
server.begin();
}
/*-----*/
void loop(){

  unsigned long actualesMilisegundos = millis();
  if (actualesMilisegundos - previosMilisegundos >= interval) {
    // Graba los últimos valores detectados por DTH
    previosMilisegundos = actualesMilisegundos;
    // Lee temperatura en Centígrados (la default)
    float newT = dht.readTemperature();

    if (isnan(newT)) {
      // Comprobamos si las lecturas de temperatura pueden dar algún fallo
      // mediante la función isnan()
      // Esta función devuelve un 1 en caso de que el valor no sea numérico
      Serial.println("Falla al leer el sensor DHT!");
    }
    else {
      t = newT;
      Serial.println(t); // Envía valor de temperatura
    }
    // lee humedad
    float newH = dht.readHumidity();
  }
}

```

```

//si la lectura de humedad read falló, , no cambies el valor de h
if (isnan(newH)) {
// Comprobamos si las lecturas de humedad pueden dar algún fallo
// mediante la función isnan()
// Esta función devuelve un 1 en caso de que el valor no sea numérico
Serial.println("Falla al leer el sensor DHT!");
}
else {
h = newH;
Serial.println(h); // Envía valor de humedad
}
}
}

```

ARCHIVO PLATFORMIO.INI

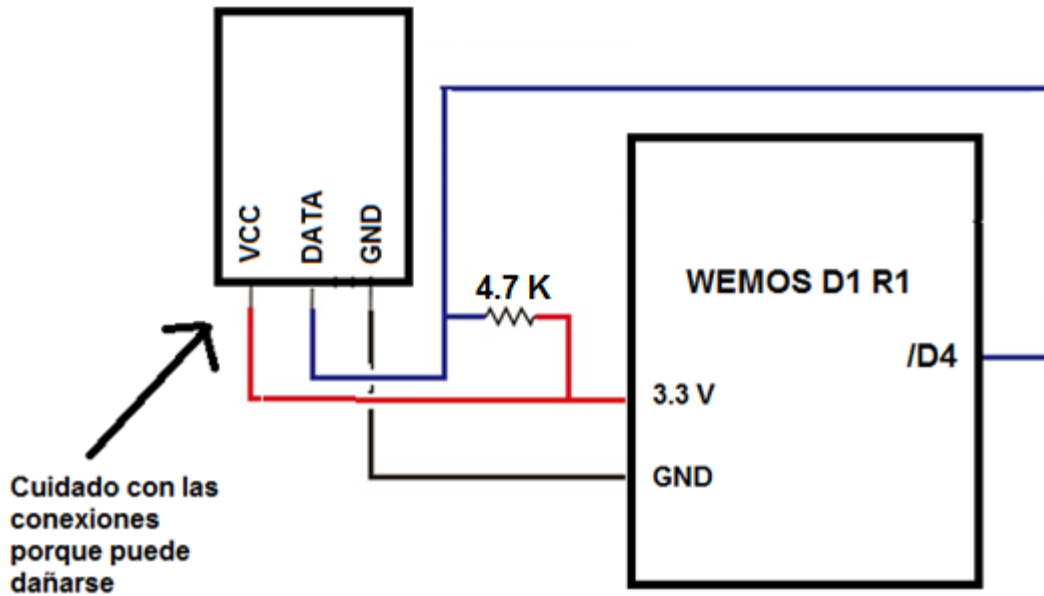
Después de instalar las librerías necesarias en este proyecto, el archivo platformio.ini debe quedar conformado de la siguiente manera:

```

[env:d1]
platform = espressif8266
board = d1
framework = arduino
lib_deps =
    ottowinter/ESPAsyncTCP-esphome@^1.2.3
    ottowinter/ESPAsyncWebServer-esphome@^1.2.7
    adafruit/Adafruit Unified Sensor@^1.1.4
    adafruit/DHT sensor library@^1.4.2
monitor_speed = 115200

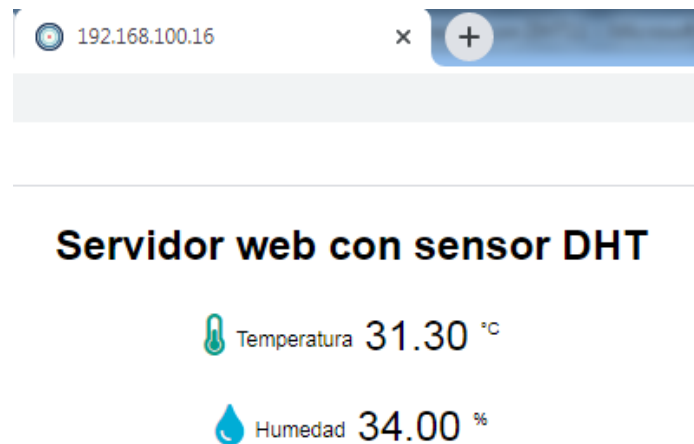
```

CONEXIONES:



PRUEBAS DEL FUNCIONAMIENTO:

Consultar la IP que proporciona el monitor de Platformio y teclear esa IP en el navegador, debe aparecer lo siguiente:

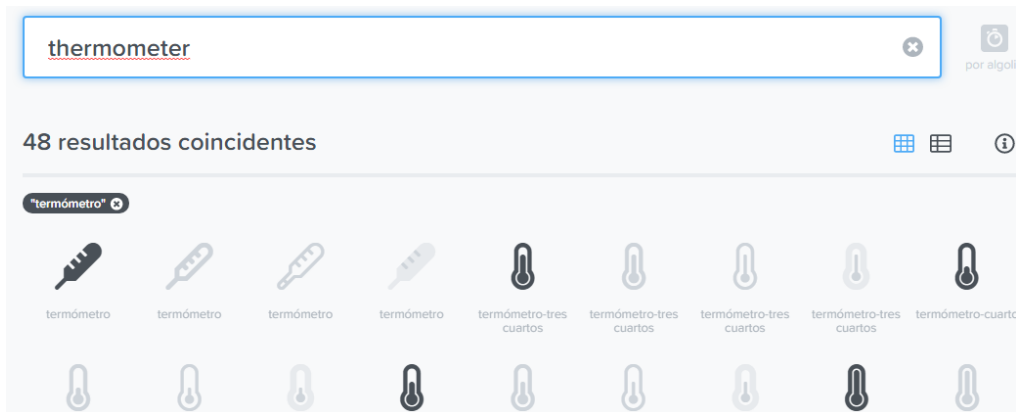


PROCEDIMIENTO PARA AGREGAR ÍCONOS A LOS DATOS MOSTRADOS:

Abrir la URL del link fuente impresionante de íconos (Font Awesome Icons).

<https://fontawesome.com/icons?d=gallery&p=2>

Selecciona una clasificación de búsqueda, por ejemplo: Thermometer



Selecciona el ícono que se desee.

Pulsar el botón 'comience a usar este ícono'. Aparecerá el código HTML

Comience a usar Font Awesome

Copie este HTML para usar este icono:

```
<i class="fas fa-thermometer-three-quarters">
```

¿No tienes un proyecto configurado? [Empiece aquí](#)