

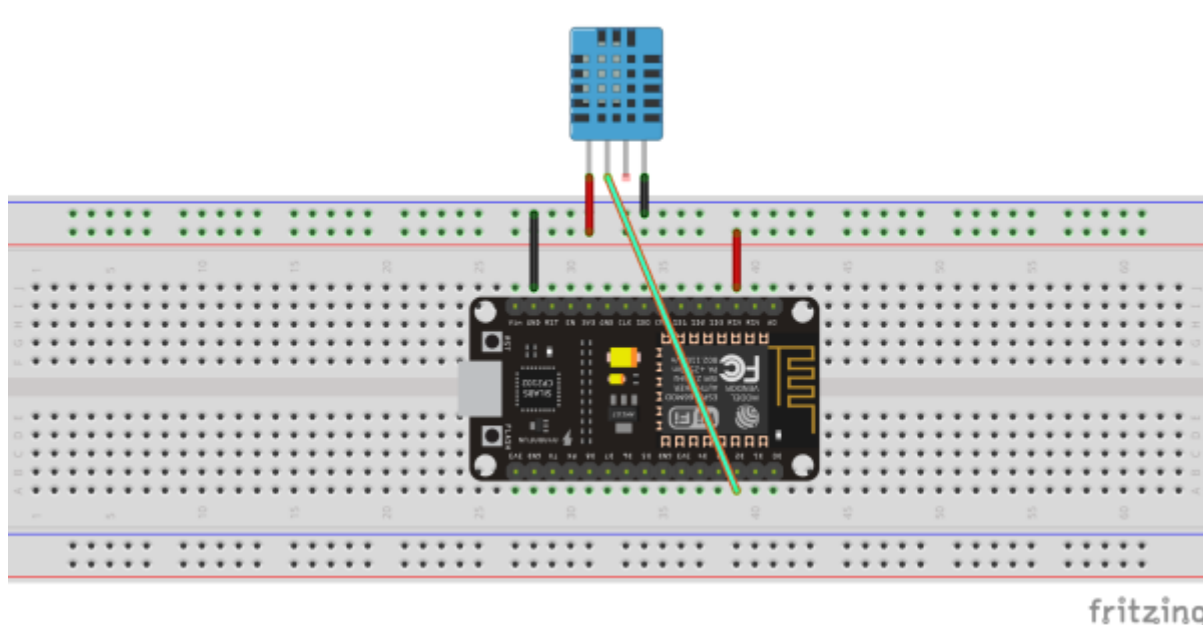
# 1. Servidor Web de humedad y temperatura

En esta práctica vamos a utilizar el ESP8266 para monitorizar via web la humedad y temperatura de un sensor DHT11. Posteriormente modificaremos el código para enviarlo a un servidor en la nube desde el que monitorizar desde cualquier punto.

## Materiales

- ESP8266-12E NodeMCU
- Sensor DHT11

## Esquema



En el esquema la patilla RSV a la que está conectado el sensor, es en la versión v3 del modeMCU la salida VU que corresponde con una salida a 5V del propio puerto usb

## Código

```
// Libreria para Sensores DHT
#include "DHT.h"

#define DHTPIN 4      // Pin del ESP8266 al que está conectado. El 4
corresponde al D2 del ESP8266
```

```
// Descomentar segun el tipo de sensor DHT usado
#define DHTTYPE DHT11 // DHT 11

// Inicializa el sensor
DHT dht(DHTPIN, DHTTYPE);

// Configura Arduino
void setup() {
    Serial.begin(115200);
    dht.begin();
}

void loop() {
    // Espera tres segundos entre mediciones. Con 2s daba problemas de lectura
    delay(3000);

    // Obtiene la Humedad
    float h = dht.readHumidity();

    // Obtiene la Temperatura en Celsius
    float t = dht.readTemperature();

    // Control de errores, valida que se obtuvieron valores para los datos medidos
    if (isnan(h) || isnan(t)) {
        Serial.println("Falla al leer el sensor DHT!");
        return;
    }

    Serial.print("Humedad: ");
    Serial.print(h);
    Serial.print(" %\t");
    Serial.print("Temperatura: ");
    Serial.print(t);
    Serial.println(" *C ");
}
```

Una vez que hemos comprobado por el serial que funciona correctamente, vamos a modificar el programa para obtener los datos desde el navegador

```
// Librerias
#include "ESP8266WiFi.h"
#include "DHT.h"

// WiFi p
const char* ssid = "intrusos";
const char* password = "12qwaszx3edc";
```

```
#define DHTPIN 4      // Pin del ESP8266 al que está conectado. El GPIO 4
corresponde al D2 del ESP8266-12E NodeMCU v3

// Descomentar segun el tipo de sensor DHT usado
#define DHTTYPE DHT11 // DHT 11

WiFiServer server(80);

// Inicializa el sensor
DHT dht(DHTPIN, DHTTYPE);

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

  // We start by connecting to a WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  // Start the server
  server.begin();
  Serial.println("Server started");
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }

  // Espera tres segundos entre mediciones
  delay(3000);

  // Obtiene la Humedad
  float h = dht.readHumidity();

  // Obtiene la Temperatura en Celsius
  float t = dht.readTemperature();
```

```
// Control de errores, valida que se obtuvieron valores para los datos
medidos
if (isnan(h) || isnan(t)) {
    Serial.println("Falla al leer el sensor DHT!");
    client.print("Falla al leer el sensor DHT");
    return;
}
// Preparamos la respuesta
// String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE
HTML>\r\n<html>\r\nMedida de Temperatura y Humedad";
String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE
HTML>\r\n";
s += "\r\n<html><head><meta http-equiv=\"refresh\"
content=\"5\"></head><body>\r\n"; // refresca automáticamente la página cada
5s
s += "<h2>Medida de Temperatura y Humedad</h2>";
s += "Humedad=";
s += (h);
s += "%<br>";
s += "Temperatura=";
s += (t);
s += "*C<br>";
s += "</body></html>\r\n";
client.print(s);
}
```

## Referencias

- <http://www.instructables.com/id/Midiendo-temperatura-con-ESP8266-y-DHT11/?ALLSTEPS>
- <http://www.instructables.com/id/Autonomous-Weather-Station-With-ESP8266/?ALLSTEPS>

From:  
<http://intrusos.info/> - LCWIKI

Permanent link:  
<http://intrusos.info/doku.php?id=electronica:esp8266:webtemperatura&rev=1472577330>

Last update: **2023/01/18 14:15**

