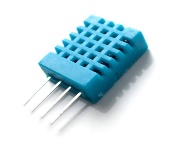
Sensor de temperatura y humedad DHT11

**Descripción general**

El sensor de temperatura DHT11 es un dispositivo electrónico utilizado para medir la temperatura y la humedad en el ambiente. Es un sensor económico y ampliamente utilizado en proyectos de electrónica, domótica y automatización.

Sensor DHT11

El DHT11 está diseñado para ser fácil de usar y proporciona datos digitales de temperatura y humedad que pueden ser leídos por microcontroladores como Arduino, Raspberry Pi y otros dispositivos similares. Este sensor utiliza un sensor capacitivo de humedad y un termistor para medir la temperatura.

**Estructura**

El DHT11 suele tener tres pines: uno para la alimentación (VCC), otro para la tierra (GND) y un tercero para la salida de datos (DATA). El sensor se comunica mediante un protocolo de una sola línea, lo que significa que los datos de temperatura y humedad se transmiten en un formato digital simple. La lectura de datos del DHT11 puede ser un poco lenta en comparación con sensores más avanzados, y su precisión también puede ser limitada en comparación con sensores más costosos.

**Descripción en detalle**

El DHT11 es un sensor digital muy usado en proyectos con Arduino, ESP32 y otras placas, porque permite medir temperatura y humedad relativa del aire de manera sencilla y económica.

Aquí te explico cómo funciona:

**1. Qué mide**

* Temperatura en grados Celsius (rango aproximado: 0 °C a 50 °C).
* Humedad relativa en % (rango: 20% a 90%).

**2. Cómo está construido**

El DHT11 está compuesto por dos partes principales:

Sensor de humedad: usa un material resistivo/capacitivo que cambia su resistencia según la cantidad de agua en el aire.

Sensor de temperatura: es un termistor que varía su resistencia con la temperatura.

Chip interno: digitaliza esas señales analógicas y las envía como datos binarios a la placa que lo controla.

**3. Comunicación con Arduino/ESP**

* El DHT11 utiliza un único pin de datos (además de VCC y GND).
* Ese pin funciona con un protocolo digital propietario (no es I2C ni SPI).

**El proceso es así:**

El microcontrolador (Arduino, ESP32, etc.) envía una señal de inicio manteniendo el pin en bajo unos milisegundo. El DHT11 responde confirmando que está listo. Luego, el sensor transmite 40 bits de información:

* 8 bits: humedad entera.
* 8 bits: humedad decimal (en DHT11 siempre es 0, ya que solo mide enteros).
* 8 bits: temperatura entera.
* 8 bits: temperatura decimal (también suele ser 0).
* 8 bits: checksum (un número de verificación para saber si los datos llegaron bien).

**4. Características principales**

* Voltaje de operación: 3.3 V – 5 V.
* Tiempo de muestreo: cada 1 segundo aprox.
* Precisión:
* Temperatura: ±2 °C.
* Humedad: ±5%.
* Muy fácil de usar gracias a librerías como DHT.h en Arduino.

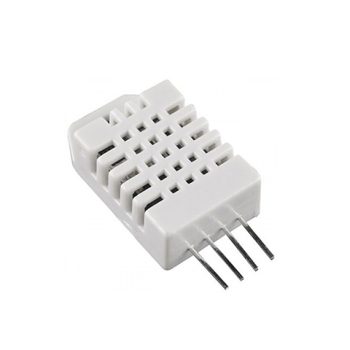
**5. Limitaciones**

* Mide solo en enteros (sin decimales).
* No es muy rápido ni muy preciso comparado con su hermano mayor, el DHT22.
* No funciona bien en ambientes extremos (alta humedad prolongada o temperaturas fuera de rango).

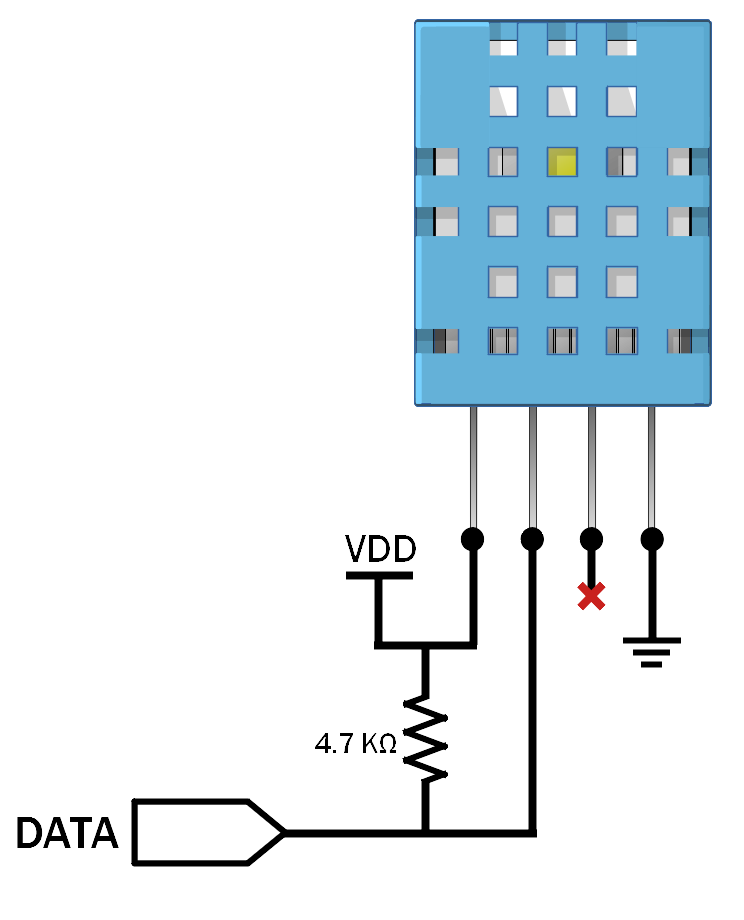
El DHT11 convierte los cambios de humedad y temperatura en señales eléctricas, las procesa con su chip interno y entrega los valores ya en formato digital mediante un solo cable de datos, lo que lo hace muy sencillo de integrar en proyectos de electrónica.

**Nota**

Es importante tener en cuenta que, si bien el DHT11 es económico y fácil de usar, no es ideal para aplicaciones que requieren alta precisión y estabilidad. En esos casos, podrían preferirse sensores más avanzados como el DHT22 o sensores de otras marcas que ofrezcan características mejoradas de medición de temperatura y humedad.

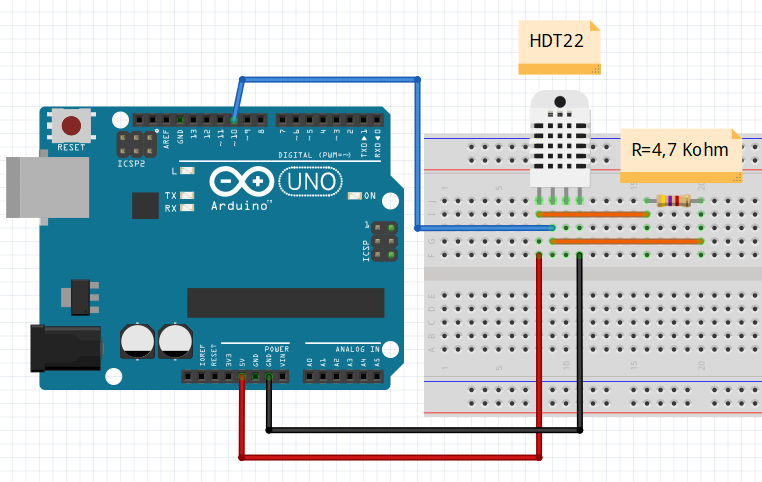


Sensor DHT22



**Terminales de conexión**

Circuito en protoboard



Programa en Arduino

#include <DHT.h>

DHT dht(10, DHT11); // O DHT22 según el sensor que usemos.

void setup()

{

  Serial.begin(9600);

  Serial.println("Prueba DHT11:");

  dht.begin();

}

void loop()

{

  // Se lee la humedad y la temperatura del sensor:

  float h = dht.readHumidity();

  float t = dht.readTemperature();

// Si los valores leídos no son números se advierte de ello, de lo contrario se imprimen los resultados:

  if(isnan(t) || isnan(h))

  {

    Serial.println("Fallo de lectura del DHT");

  }

  Else

  {

    Serial.print("Humedad: ");     Serial.print(h); Serial.print(" %\t");

    Serial.print("Temperatura: "); Serial.print(t); Serial.print(" \*C\n");

  }

  delay(1000);

}

Ejercicio

Hacer un programa que active un led cuando la temperatura supere un valor de referencia adecuado y se apague al bajar a valores menores al de referencia.

Hacer un programa que active una sirena si la temperatura supera un determinado valor de referencia.

Espacio para próximos ejercicios.