

Características del sensor dht11

Se recomienda solo leer
la información sobre DHT11



Contenido

[Características principales del sensor dht11.](#)

[Como conectar el sensor dht11 al circuito NodeMcu](#)

[Como programar el sensor dht11 en Lua en NodeMCU](#)

[Como programar el sensor dht11 en Arduino](#)

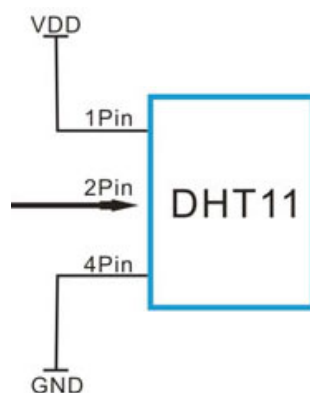
Características principales del sensor dht11.

Este sensor de humedad y temperatura es de D-Robotics y con el puedes obtener un muy buen resultado, en mi opinión.

Con este sensor dht11 consigues tener una medida de la temperatura y humedad muy exacta donde se encuentre, con una respuesta rápida y sin interferencias.

Es un sensor resistente, sencillo y muy practico con un precio muy económico.

El **sensor dht11** dispone de 3 pins, un pin para la entrada de tensión, otro pin para la masa, y otro pin para la transferencia de datos.

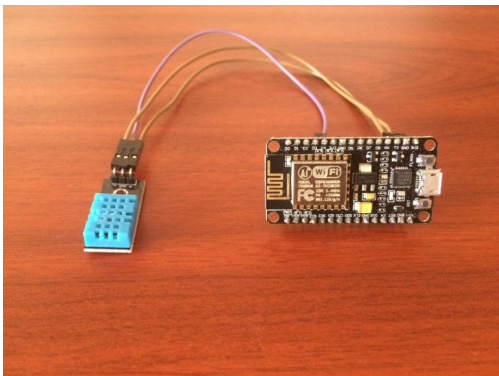


Es capaz de transmitir la información hasta 20 metros de distancia, la tensión de alimentación es de 3 a 5.5 Voltios en continua, con una corriente de trabajo de 2.5 miliamperios.

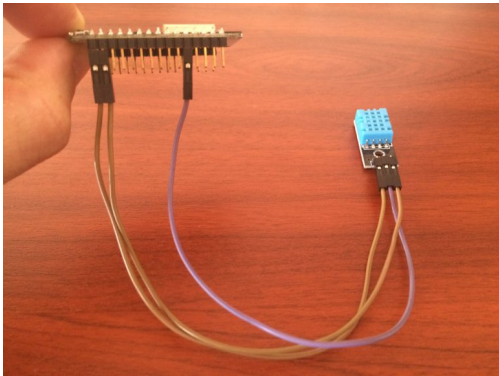
Con este sensor podemos medir temperaturas entre 0 y 50 grados Centígrados con un error de mas menos un 2%, y en cuanto a la humedad podemos medir el porcentaje de humedad con un rango entre 20 y 90% de humedad y un margen de error de un mas menos 5%.

Como conectar el sensor dht11 al circuito NodeMcu

Para ver la lectura de temperatura y humedad del sensor dht11 vamos a utilizar el circuito impreso [nodemcu](#).

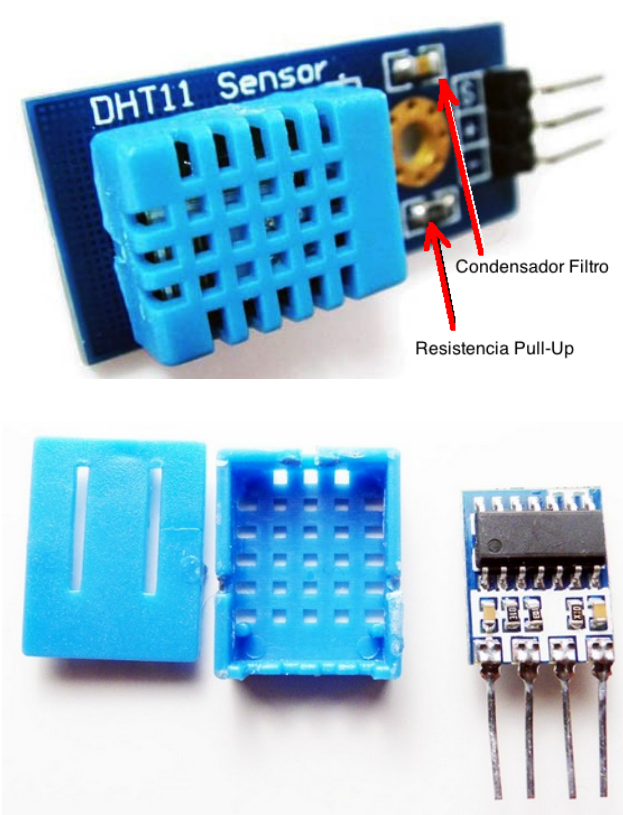


Conectaremos el sensor dht11 con los pines de los extremos para dar tensión al sensor a los pines de tensión de la nodemcu, que son los cables marrones, y el pin de datos que esta en el medio de los pines del sensor irá conectado al pin "D4", entrada de señal digital del nodemcu, es el cable morado.

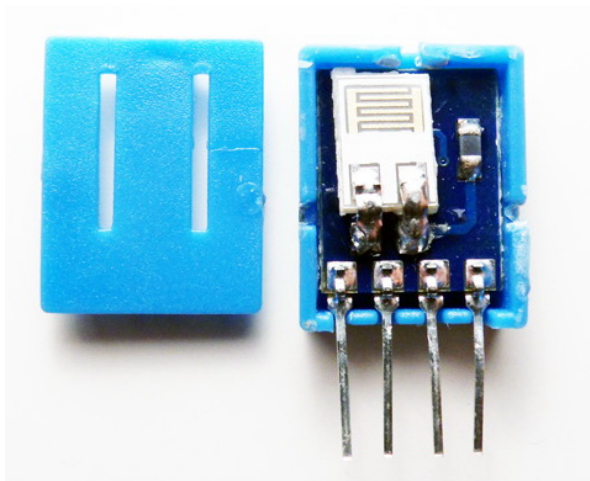


En la imagen de abajo podemos ver la composición externa al propio sensor dht11 que incluye esta versión del sensor. En la placa tenemos un condensador de filtro diseñado para rectificar las fluctuaciones de tensión en la alimentación ya sea 3.3 voltios o 5 voltios.

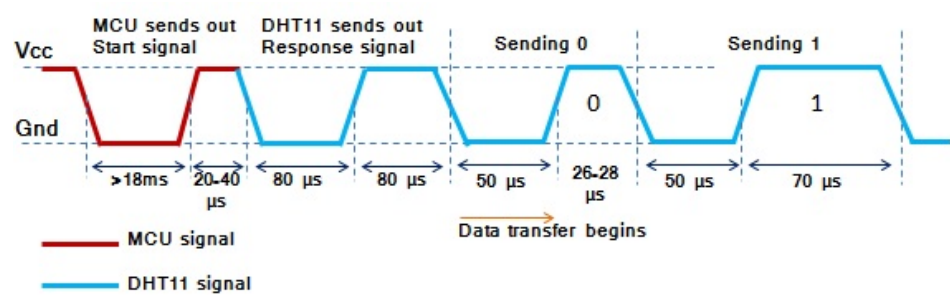
También tenemos una resistencia pull-up diseñada para evitar el ruido o las medidas erróneas de la señal enviadas al MCU o en nuestro caso a la NODEMCU.



En la imagen de arriba vemos destripado el sensor DHT11 donde se puede apreciar un conjunto de resistencias abajo y arriba la memoria OTP (one-time-programmable) donde quedan grabados los coeficientes de medición, calculados en el proceso de calibración del sensor.



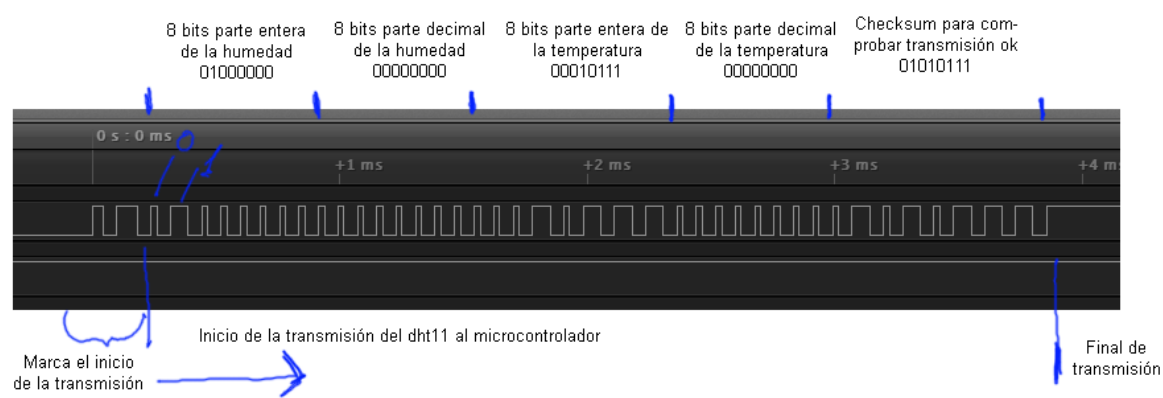
En la imagen de arriba vemos los dos sensores de temperatura y el de humedad. El sensor de temperatura es de tipo NTC (mide la temperatura a través de una resistencia) y el sensor de humedad HR202.



En la imagen de arriba vemos el protocolo para el envío de datos del sensor DHT11.

La señal se compone de **40 bits** transferidos, estos 40 bits transferidos se componen de **8 bits** para indicar el valor de la **humedad con formato entero**, más **8 bits** para transmitir la señal de la **humedad con un formato decimal**, luego **8 bits** más para transmitir **la temperatura en formato entero** más **8 bits** para enviar la señal de **temperatura en formato decimal**, y los **últimos 8 bits** son para enviar **un check sum** que verifica la información anterior por si hubiera errores.

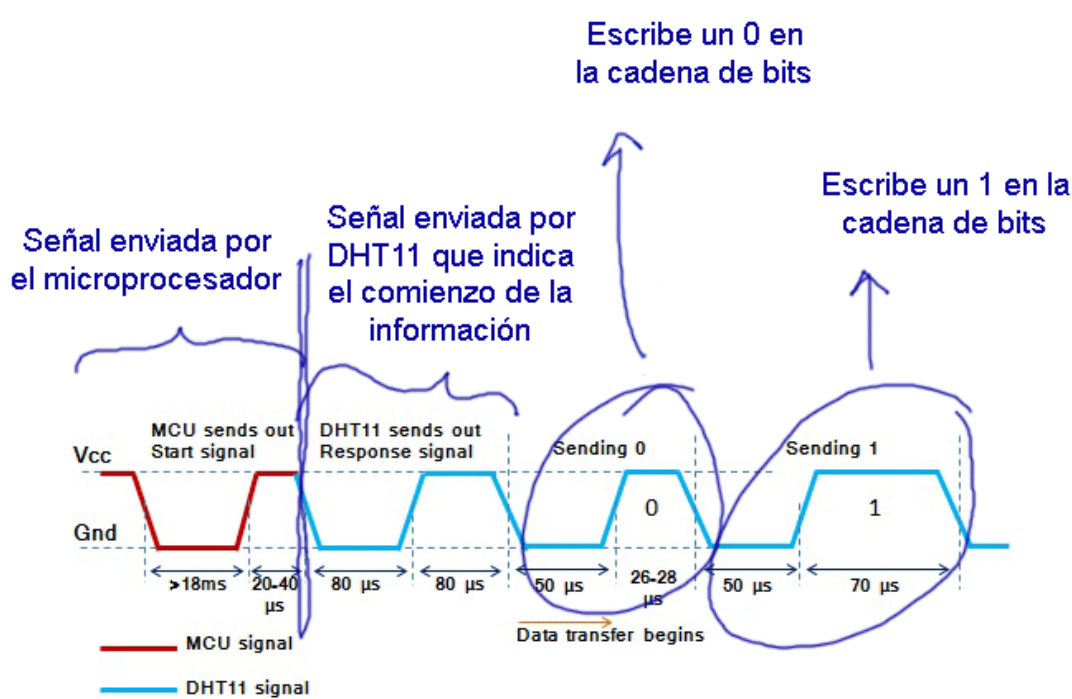
El diagrama con una humedad de 64 % y 23 grados centígrados es la siguiente imagen:



En la imagen de arriba tenemos representada la serie de ceros y unos correspondientes a los estados bajos y altos que emite el dht11 para comunicar al microcontrolador la temperatura y la humedad que esta detectando.

Para representar un "cero" digital el dht11 envía 50 microsegundos de estado bajo, cero tensión, seguidos de un estado alto con 5,5 o 3,5 voltios, con una duración de 26 a 28 microsegundos.

Mientras que para representar el "uno" digital el dht11 envía o transmite 50 microsegundos de estado bajo seguidos de 70 microsegundos de estado alto.



Los dos primeros bits indican el inicio de la transmisión de datos, los 8 bits siguientes representan el valor de la humedad en binario, los siguientes 8 bits representan el valor decimal de la humedad, los siguientes 8 bits representan el valor de la temperatura y los 8 siguientes bits representan el valor decimal de la temperatura.

Los últimos 8 bits se utilizan para verificar que los datos tienen una relación entre si, y no se ha producido ningún error de paridad. Los últimos 8 bits son una suma de los 32 bits anteriores:

$$01000000+00000000+00010111+00000000=01010111$$

Ejemplo 1:

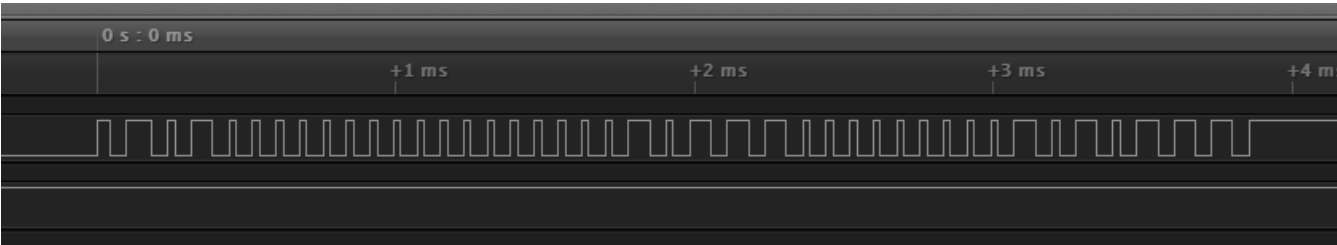
8 bits de dato integral de T = 0x01 más 8 bits de dato decimal de T = 0x32

El resultado de la temperatura medida es:

$$((0x01 \ll 8) + 0x32) / 10 = (256 + 50) / 10 = 30.6 \text{ grados centígrados.}$$

Ejemplo 2:

Una muestra del resultado de la transferencia de datos seria la de la imagen:



En este ejemplo podemos ver los bits que dan la información de inicio propia de las señales asincronas y al final los bits que dan la señal de final de transmisión.

Esta información esta compuesta por 4 bloques de 8 bits.

Los 8 primeros bits que dan la información de la humedad serían "01000000" que en decimal sería 64 , que quiere decir 64 % de humedad.

El tercer bloque de 8 bits nos da la información de la temperatura que en este caso es: "00010111" en decimal es el número 23, correspondiente a 23 grados centígrados.

Como programar el sensor dht11 en Lua en NodeMCU - Lua es lenguaje de programacion

Después deberemos de conectar la nodemcu al puerto usb del ordenador con el que vayamos a programar el circuito con el sensor. Deberemos [flashear](#) la nodemcu e instalarle el firmware con los drivers del sensor dht11. Para instalar el firmware que necesitamos podemos construirlo nosotros mismos desde la pagina "<http://nodemcu-build.com/>".

Para descargar el programa en la nodemcu podemos utilizar "ESPlorer", el programa tendría las siguientes líneas:

```
1 //Programación con nodemcu
2 in = 4 --corresponde al D4
3 humi=0
4 temp=0
5
6 --Funcion leer sensor DHT11
7 function LeerDHT11()
8 status,temp,humi=dht.read11(pin)
9
10 if(status==dht.OK) then
11 print("DHT Temperatura: "..temp.."";.."Humedad: "..humi)
12 elseif(status==dht.ERROR_CHECKSUM)then
13 print("DHT Checksum error.");
14 elseif(status==dht.ERROR_TIMEOUT)then
15 print("DHT Time out.");
16 end
17 end
18 LeerDHT11()
```

De esta manera y con este codigo, cada vez que hagamos un "dofile" en "init.lua" clikeando sobre el nombre del archivo que tenemos a la derecha de "ESPlorer", podemos ver el archivo despues de descargarlo, con el botón de "Reload" de la derecha.

Como programar el sensor dht11 en Arduino

Utilizando el entorno de programación de Arduino el programa quedaría de la siguiente manera:

```
1 //Programación con arduino
2 #include "DHT.h"
3
4 #define DHTTYPE DHT11 // DHT 11
5 // DHT Sensor
6 const int DHTPin = 2; //el número 2 corresponde con el pin D4 en ESP12, para el ESP32 el D4 es el 4
7 //otra opción sería definir una constante para el pin D4 por ejemplo #define DHTPin D4
8
9 DHT dht(DHTPin, DHTTYPE);
10
11 void setup() {
12
13     Serial.begin(115200);
14
15     dht.begin();
16
17 }
18 void loop() {
19
20     float h = dht.readHumidity();
21
22     float t = dht.readTemperature();
23
24     Serial.print("DHT Temperatura: ");
25     Serial.print(t);
26     Serial.print(" °C\t Humedad: ");
27     Serial.print(h);
28     Serial.println(" % ");
29     delay(500);
30 }
```