

**Parte 2**

# Lección

# 10

**Sensor de humedad  
y temperatura DHT11**

## Resumen

En este tutorial vamos a aprender cómo usar un Sensor de humedad y temperatura DHT11.

Es lo suficientemente exacta para la mayoría de los proyectos que necesitan hacer un seguimiento de las lecturas de humedad y temperatura.

Otra vez vamos a usar una biblioteca diseñada específicamente para estos sensores que harán que nuestro código corto y fácil de escribir.

### Componente necesario:

(1) x Elegoo Uno R3

(1) x módulo de humedad y temperatura DHT11

(4) x F M cables (cables de hembra a macho DuPont)

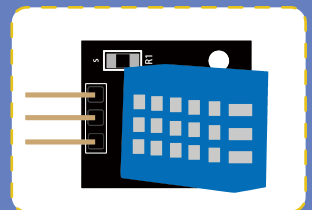


## Introducción del componente

### Sensor de temperatura y humedad:

**Sensor** digital de temperatura y humedad DHT11 es un Sensor compuesto que contiene la salida de la señal digital calibrado de la temperatura y la humedad. La tecnología de colección de módulos digitales dedicado y la temperatura y humedad sensor de tecnología se aplican para garantizar que el producto tiene alta confiabilidad y excelente estabilidad a largo plazo. El sensor incluye un sentido resistente de componentes mojados y un dispositivos de medición de temperatura NTC y conecta con un microcontrolador de 8 bits de alto rendimiento.

**Aplicaciones:** HVAC, deshumidificador, ensayos e inspección de equipos, bienes de consumo, control automático, automóvil, registradores de datos, estaciones meteorológicas, electrodomésticos, regulador de humedad, humedad médicos y otros medición y control.



### Parámetros del producto

Humedad relativa:

1m / s air 6s

Resolución: 8Bit

Histéresis:  $\leq \pm 0.3\% \text{ RH}$

Repetibilidad:  $\pm 1\% \text{ RH}$

Estabilidad a largo plazo:  $\leq \pm 0.5\% \text{ RH / yr in}$

Precisión: At  $25^{\circ}\text{C} \pm 5\% \text{ RH}$

Intercambiabilidad: fully interchangeable

Tiempo de respuesta: 1 / e (63%) of  $25^{\circ}\text{C}$  6s

Tiempo de respuesta: 1 / e (63%) 10S

Temperatura:

Características eléctricas

Resolución: 8Bit

Fuente de alimentación: DC 3.5~5.5V

Repetibilidad:  $\pm 0.2^{\circ}\text{C}$

Corriente: medición 0.3mA espera 60μA

Rango: At  $0^{\circ}\text{C} \pm 50^{\circ}\text{C}$

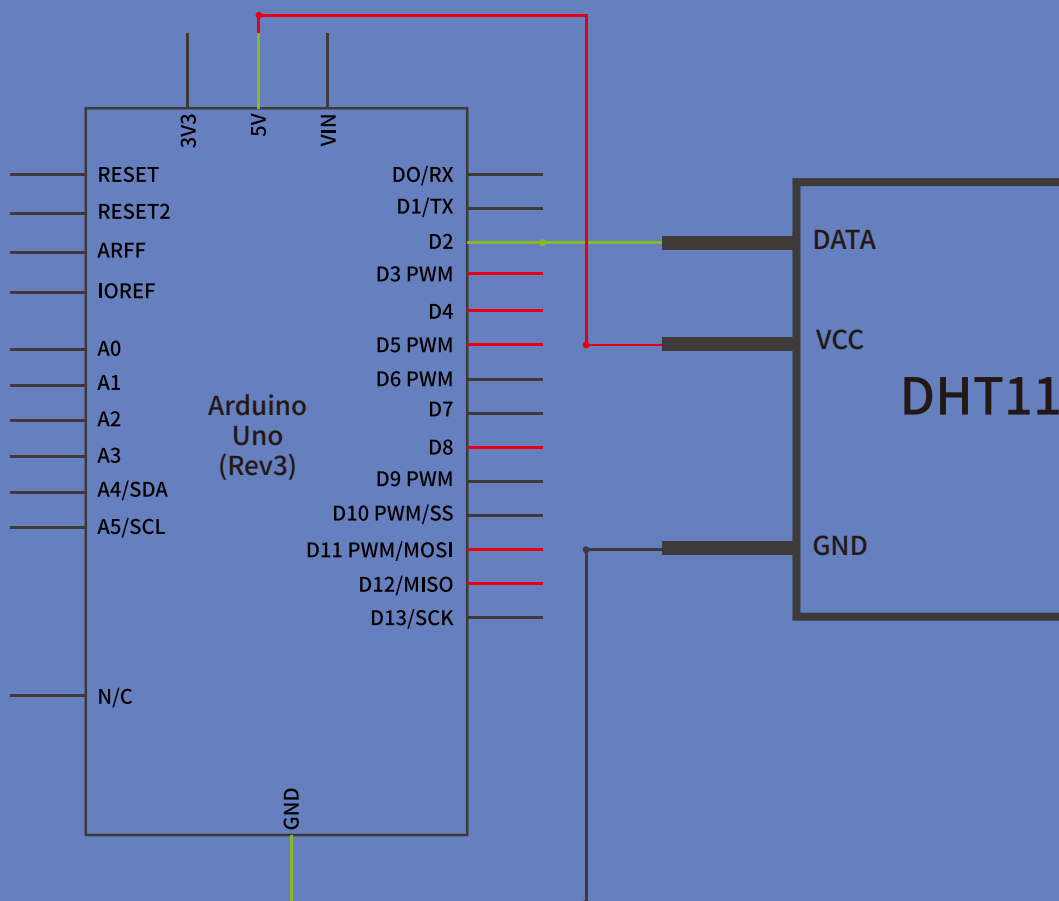
Periodo de muestreo: más de 2 segundos

### PIN Descripción:

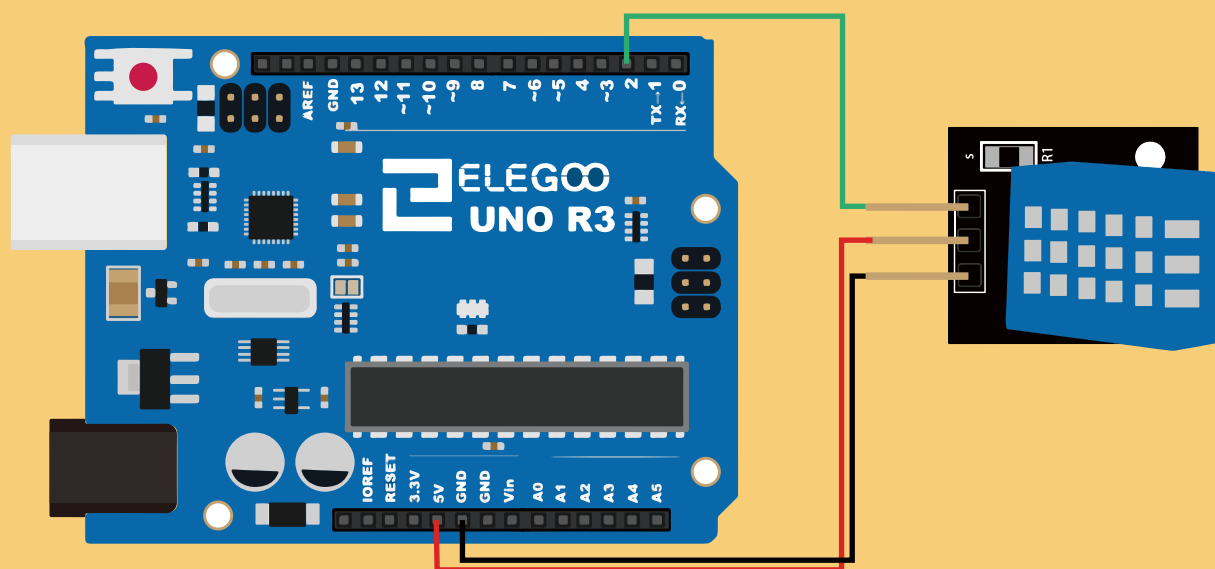
1.VDD alimentación 3,5~5.5V DC

2.serie de datos, un solo bus

3.GND tierra, la energía negativa



## Conexión Esquema



Como se puede ver que sólo necesitamos 3 conexiones al sensor, ya que uno de lo pin no se utiliza. Las conexiones son: voltaje, tierra y señal de que puede conectarse a cualquier Pin en nuestro UNO.

## Diagrama de cableado

## Code

Después del montaje, abra el programa en la carpeta “DHT” donde se encuentra el curso y haga clic en CARGAR para cargar el programa. Consulte la Lección 5 en la parte 1 para obtener detalles sobre la carga del programa si hay algún error.

### static

#### [Alcance de variable y calificadores]

##### Descripción

La palabra clave “static” se usa para crear variables que son visibles para una sola función. Sin embargo, a diferencia de las variables locales que se crean y eliminan cada vez que se llama a una función, las variables estáticas persisten más allá de la llamada a la función, preservando sus datos entre llamadas a funciones.

Las variables declaradas como estáticas solo se crearán e inicializarán la primera vez que se llame a una función.

### float

#### [Tipos de datos]

##### Descripción

Tipo de datos para números de punto flotante, un número que tiene un punto decimal. Los números de punto flotante a menudo se usan para aproximar valores analógicos y continuos porque tienen una resolución mayor que los enteros. Los números de coma flotante pueden ser tan grandes como 3.4028235E+38 y tan bajos como -3.4028235E+38. Su almacenamiento consume 32 bits (4 bytes) de información.

##### Sintaxis

**float** var = val;

##### Parámetros

**var:** nombre de la variable.

**val:** Valor que se asigna a esa variable.

```
static const int DHT_SENSOR_PIN = 2;
```

```
float temperature;
```

```
float humidity;
```

## Bool

#### [Tipos de datos]

##### Descripción

Un “bool” contiene uno de los dos valores, verdadero o falso. (Cada variable bool ocupa un byte de memoria).

##### Sintaxis

**bool** var = val;

##### Parámetros

**var:** nombre de la variable.

**val:** Valor que se asigna a esa variable.

### unsigned long

#### [Conversion]

##### Description

Converts a value to the unsigned long data type.

##### Parámetros

**x:** un valor de cualquier tipo

##### Devuelve

unsigned long

```
static bool measure_environment( float *temperature, float *humidity )
{
    static unsigned long measurement_timestamp = millis( );

    /* Measure once every four seconds. */
    if( millis( ) - measurement_timestamp > 3000ul )
    {
        if( dht_sensor.measure( temperature, humidity ) == true )
        {
            measurement_timestamp = millis( );
            return( true );
        }
    }

    return( false );
}
```

- Sube el programa y enciende el monitor para ver los siguientes datos:  
Muestra la temperatura del ambiente. Podemos ver que es de 27 a 25 grados, 45,0% de humedad.
- Haga clic en el botón Serial Monitor para encender el monitor serie. Se introducen los conceptos básicos sobre el monitor serial en detalles en la lección 4 parte 2.

