



S/. 0,00

CATEGORÍAS

[All Blog News](#)[Tutorial sensor digital de temperatura DS18B20](#)

ARTÍCULOS POPULARES

[Tutorial transmisor de celda de carga HX711, Balanza Digital ago 23, 2015](#)[Configuración del módulo bluetooth HC-05 usando comandos AT ago 08, 2015](#)[Tutorial sensor de flujo de agua abr 13, 2016](#)

ARTÍCULOS RECIENTES

[Usando ESP8266 con el IDE de Arduino oct 21, 2016](#)[Tutorial Sensor de Distancia SHARP sep 28, 2016](#)



Arduino y ESP8266 como cliente web ago 24, 2016

PROMOCIONES ESPECIALES



Módulo RF 433MHz TX y RX
Controla tus proyectos de...

S/. 7,00 ~~S/. 8,00~~

Todas las promociones especiales >

ETIQUETAS



sensor

cnc

robot

Servo

LED

LCD

infrarrojo

driver

Fuente

cable

NOVEDADES



Shield GSM/GPRS Geeetech v2

La manera más sencilla y rápida de comunicar tu Arduino con la red...

S/. 170,00



Módulo Step up down XL6009 1.5A

Aumenta o disminuye el voltaje de forma eficiente

S/. 12,00



Teensy 3.6

La tarjeta de desarrollo más avanzada de la familia Teensy, mucho más...

S/. 200,00

**Rodamiento Lineal con Soporte largo D8mm - SC8LUU**

Rodamiento SC8LUU con sistema de fijación directa. Ofrece una mayor...

S/. 30,00

**Rodamiento Lineal Largo D8mm - LM8LUU**

Rodamiento lineal de D8mm y L40mm

S/. 15,00

**Kit de 10 brocas PCB**

Para prototipado de PCBs

S/. 30,00

**Spindle de 200W con abrazadera**

Ideal para Fresadoras CNC

S/. 300,00

**Collet ER11 de 3.175mm**

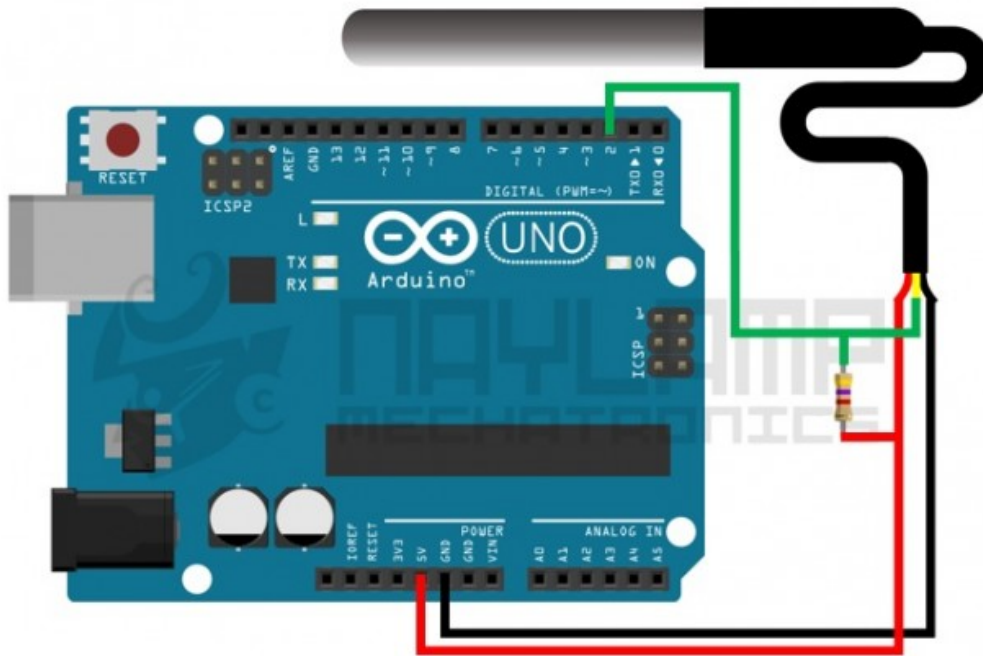
Ideal para prototipado de PCBs

S/. 20,00

[Todas los nuevos productos >](#)

Tutorial sensor digital de temperatura DS18B20

Posted by Tutoriales 14 Comments



Sensor DS18B20

El DS18B20 es un sensor digital de temperatura que utiliza el protocolo 1-Wire para comunicarse, este protocolo necesita solo un pin de datos para comunicarse y permite conectar más de un sensor en el mismo bus.

El sensor DS18B20 es fabricado por Maxim Integrated, el encapsulado de fabrica es tipo TO-92 similar al empleado en transistores pequeños. La presentación comercial más utilizada por conveniencia y robustez es la del sensor dentro de un tubo de acero inoxidable resistente al agua, con el que trabajemos este tutorial.



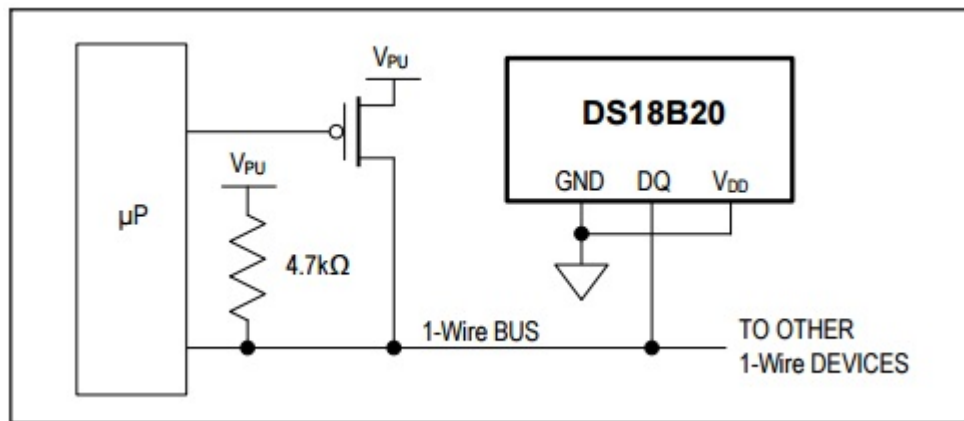
Con este sensor podemos medir temperatura desde los -55°C hasta los 125°C y con una resolución programable desde 9 bits hasta 12 bits.

Cada sensor tiene una dirección única de 64bits establecida de fábrica, esta dirección sirve para identificar al dispositivo con el que se está comunicando, puesto que en un bus 1-wire pueden existir más de un dispositivo.

El sensor tiene dos métodos de alimentación:

Alimentación a través del pin de datos:

De esta forma, el sensor internamente obtiene energía del pin de datos cuando este se encuentra en un estado alto y almacena carga en un condensador para cuando la línea de datos esté en un estado bajo, a esta forma de obtener energía se le llama "Parasite Power" y se usa cuando el sensor debe conectarse a grandes distancias o en donde el espacio es limitado, puesto que de esta forma no se necesita la línea de VDD. El diagrama para su conexión debe ser de la siguiente forma:

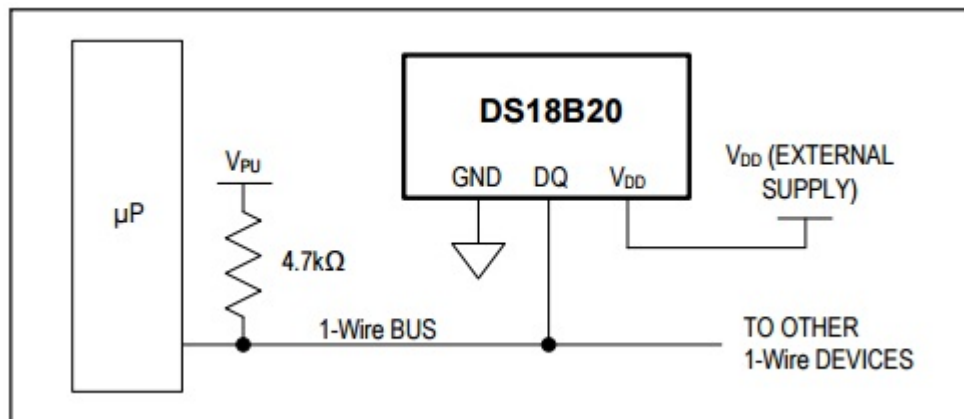


Notar que el pin GND y VDD están ambos conectados a GND, esto es indispensable para que se active el Parasite Power. EL MOSFET en la imagen es necesario para cuando se realicen conversiones de temperatura o copiar datos desde la memoria de circuito de la EEPROM, en estas operaciones la corriente de operación aumenta y si solo se suministra energía a través de la resistencia pueden causar caídas de voltaje en el condensador interno.

Alimentación usando una fuente externa:

De esta forma el sensor se alimenta a través del pin VDD, de esta forma el voltaje es estable e independiente del tráfico del bus 1-wire.

El diagrama de conexión es de la siguiente forma:



Esta forma de alimentación es la más recomendada y es la utilizada en este tutorial.

Librerías para el DS18B20 en Arduino

Para poder trabajar el DS18B20 en Arduino necesitamos dos librerías:

- Librería **OneWire**, descarga: <https://github.com/PaulStoffregen/OneWire>

En esta librería está implementado todo el protocolo del bus 1-wire. Y puede usarse tanto para el DS18B20 como para otros dispositivos 1-wire, para mayor información sobre la

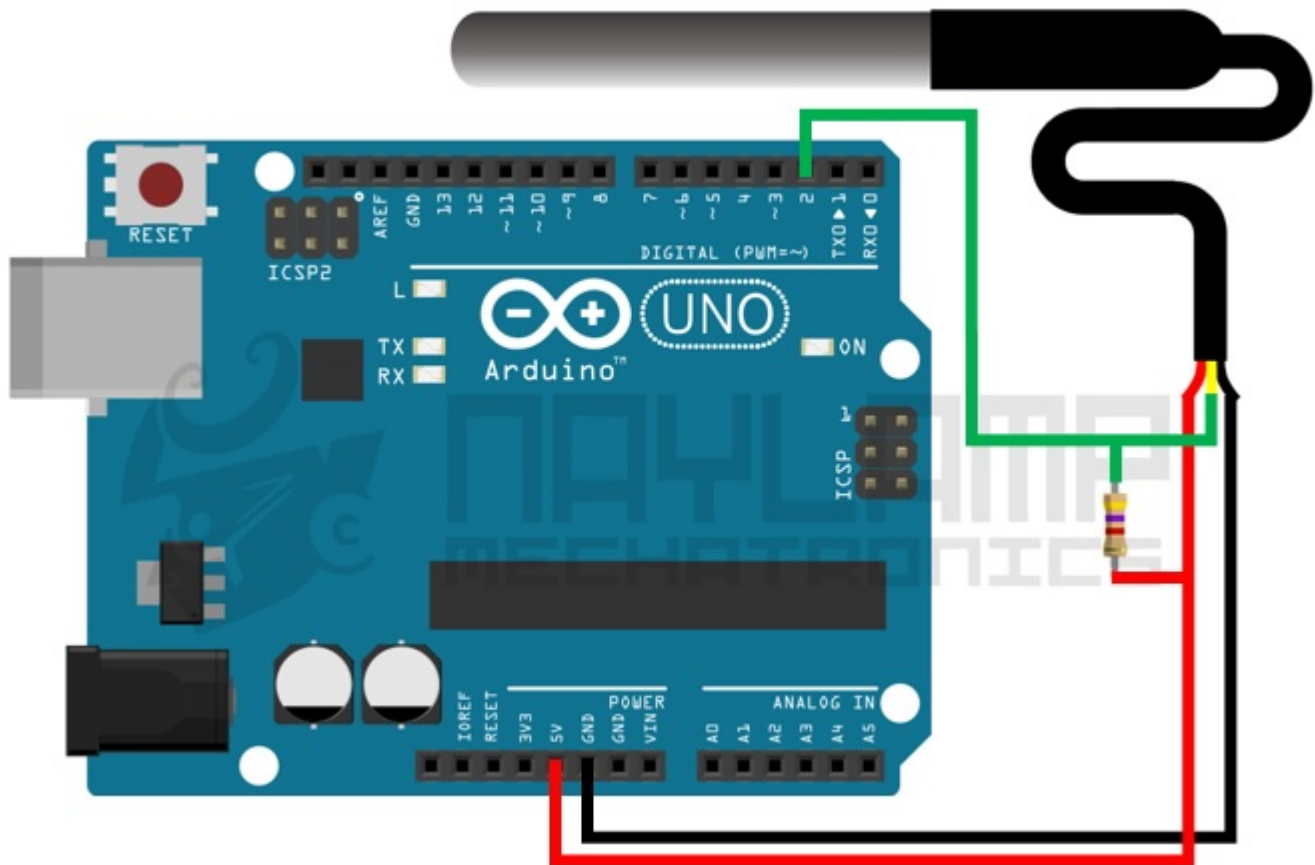
librería: http://www.pjrc.com/teensy/td_libs_OneWire.html

-Librería **DallasTemperature**, descarga: <https://github.com/milesburton/Arduino-Temperature-Control-Library>

En esta librería están implementadas las funciones necesarias para poder realizar las lecturas o configuraciones del DS18B20, para mayor información de la librería revisar:

http://www.milesburton.com/Dallas_Temperature_Control_Library

Conexiones del DS18B20 con Arduino:



Notar que es necesario poner una resistencia Pull-Up de 4.7K, y pueden usar un protoboard para facilitar la conexión.

Ej.1: Realizando lecturas de temperatura con el DS18B20

Después de instalar las librerías y realizar la conexión anterior podemos realizar las lecturas de temperatura, para eso usamos el siguiente sketch:

```
#include <OneWire.h>
#include <DallasTemperature.h>
```

```
OneWire ourWire(2); //Se establece el pin 2 como bus OneWire

DallasTemperature sensors(&ourWire); //Se declara una variable u objeto para nuestro sensor

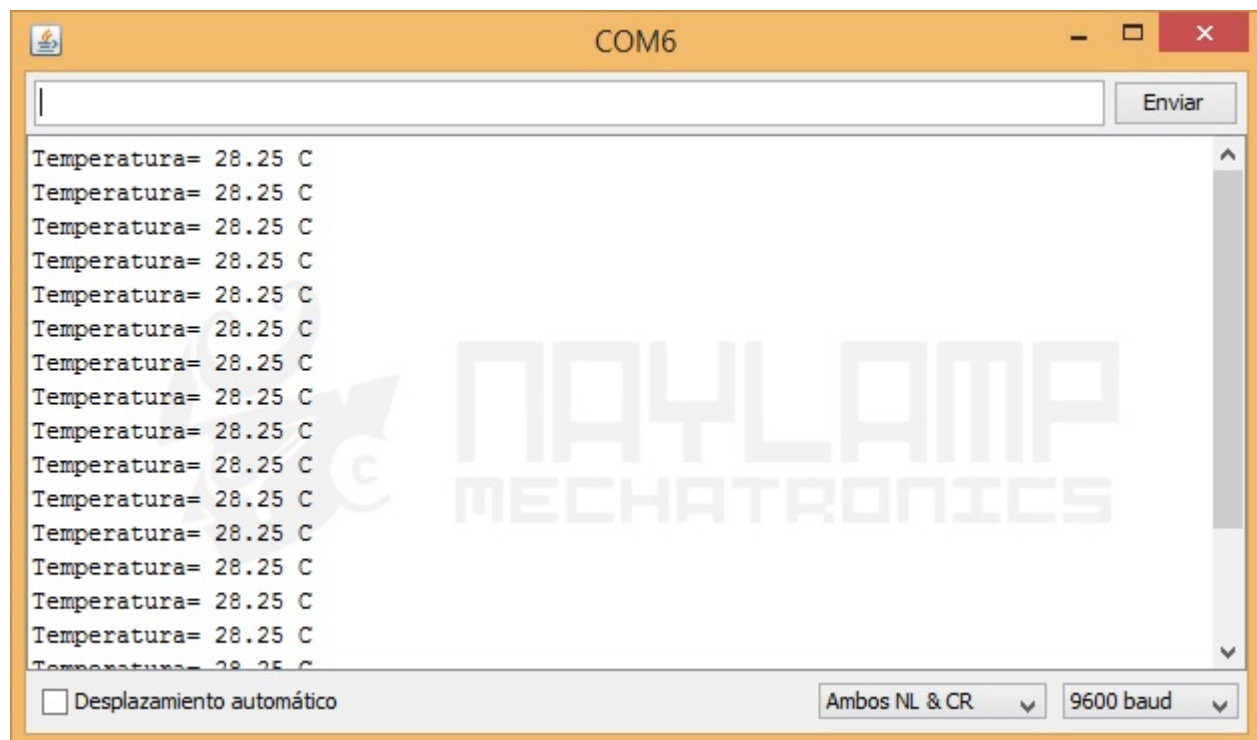
void setup() {
  delay(1000);
  Serial.begin(9600);
  sensors.begin(); //Se inicia el sensor
}

void loop() {
  sensors.requestTemperatures(); //Se envía el comando para leer la temperatura
  float temp= sensors.getTempCByIndex(0); //Se obtiene la temperatura en °C

  Serial.print("Temperatura= ");
  Serial.print(temp);
  Serial.println(" C");
  delay(100);
}
```

Como se observa medir la temperatura es simple, son necesarias solo dos líneas en el void loop() para realizar esta tarea.

El resultado es el siguiente:



Conectando varios sensores de temperatura:

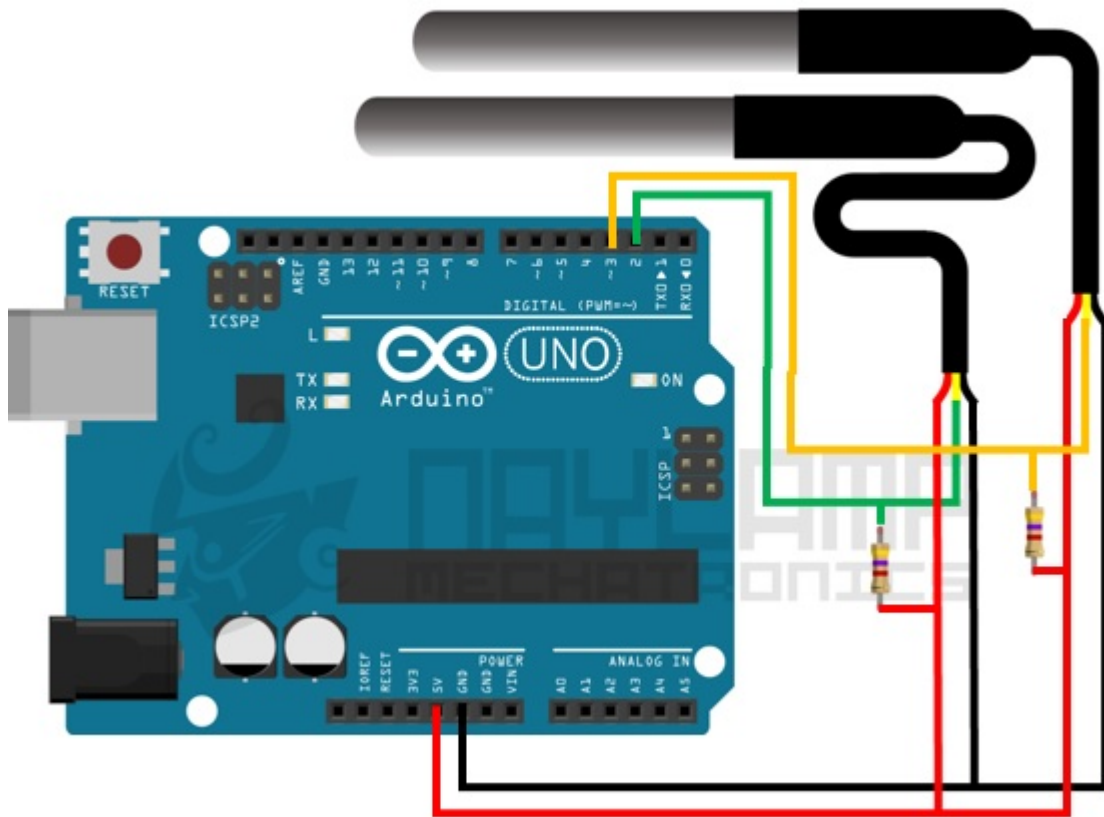
Tenemos dos opciones o métodos que podemos usar cuando necesitamos leer más de un sensor de temperatura.

El primer método es manejando cada sensor con un pin diferente del Arduino. De esta forma si tenemos 3 sensores, necesitaremos usar 3 pines digitales del Arduino.

Otra forma es usar el mismo pin para todos los sensores, en otras palabras todos los sensores se conectan al mismo bus 1-Wire, y como cualquier bus, cada elemento o dispositivo posee una identificación o dirección. En el caso del DS18B20 necesitamos averiguar su dirección que es única y viene establecida de fábrica.

Ej.2: Usando varios DS18B20 en diferentes pines del Arduino:

Para este caso las conexiones son las siguientes:



Cada sensor trabaja con un pin diferente y necesita su propia resistencia Pull-Up de 4.7K.

El código para realizar las lecturas es el siguiente:

```
#include <OneWire.h>
#include <DallasTemperature.h>

OneWire ourWire1(2);           //Se establece el pin 2 como bus OneWire
OneWire ourWire2(3);           //Se establece el pin 3 como bus OneWire

DallasTemperature sensors1(&ourWire1); //Se declara una variable u objeto para nuestro sensor1
DallasTemperature sensors2(&ourWire2); //Se declara una variable u objeto para nuestro sensor2

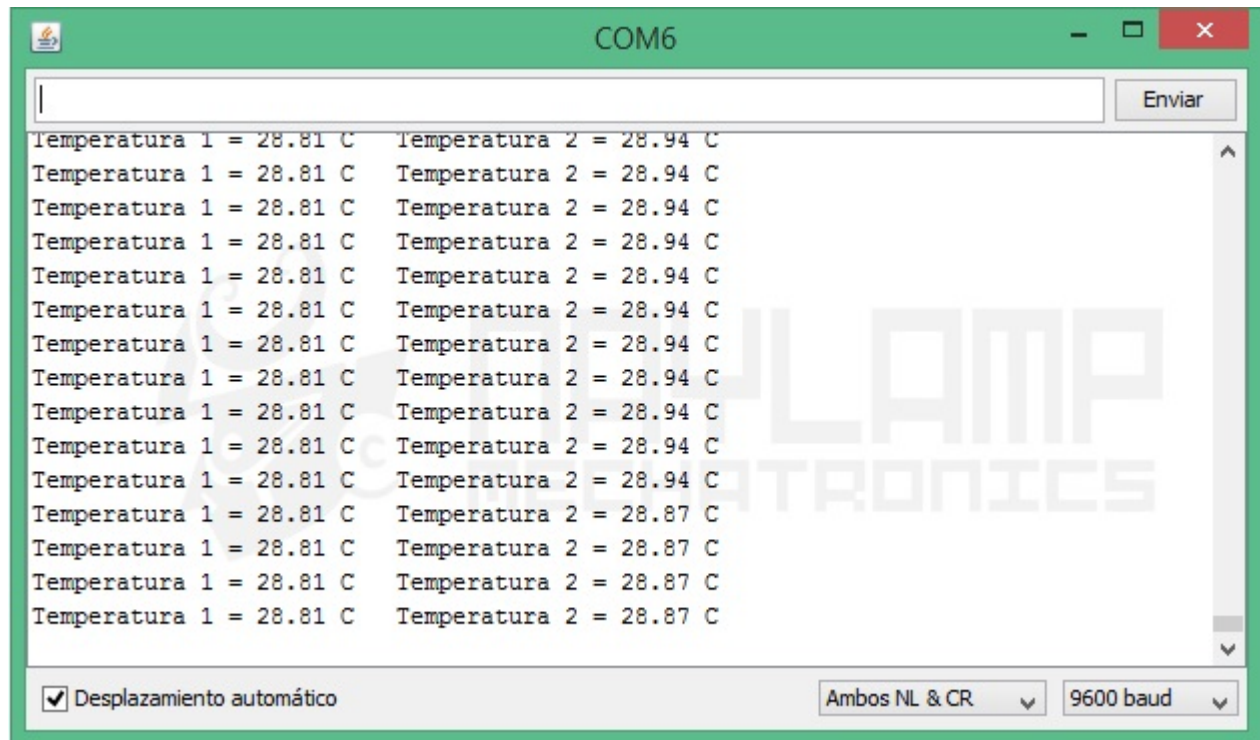
void setup() {
```

```
delay(1000);
Serial.begin(9600);
sensors1.begin(); //Se inicia el sensor 1
sensors2.begin(); //Se inicia el sensor 2
}

void loop() {
sensors1.requestTemperatures(); //Se envía el comando para leer la temperatura
float temp1= sensors1.getTempCByIndex(0); //Se obtiene la temperatura en °C del sensor 1

sensors2.requestTemperatures(); //Se envía el comando para leer la temperatura
float temp2= sensors2.getTempCByIndex(0); //Se obtiene la temperatura en °C del sensor 2

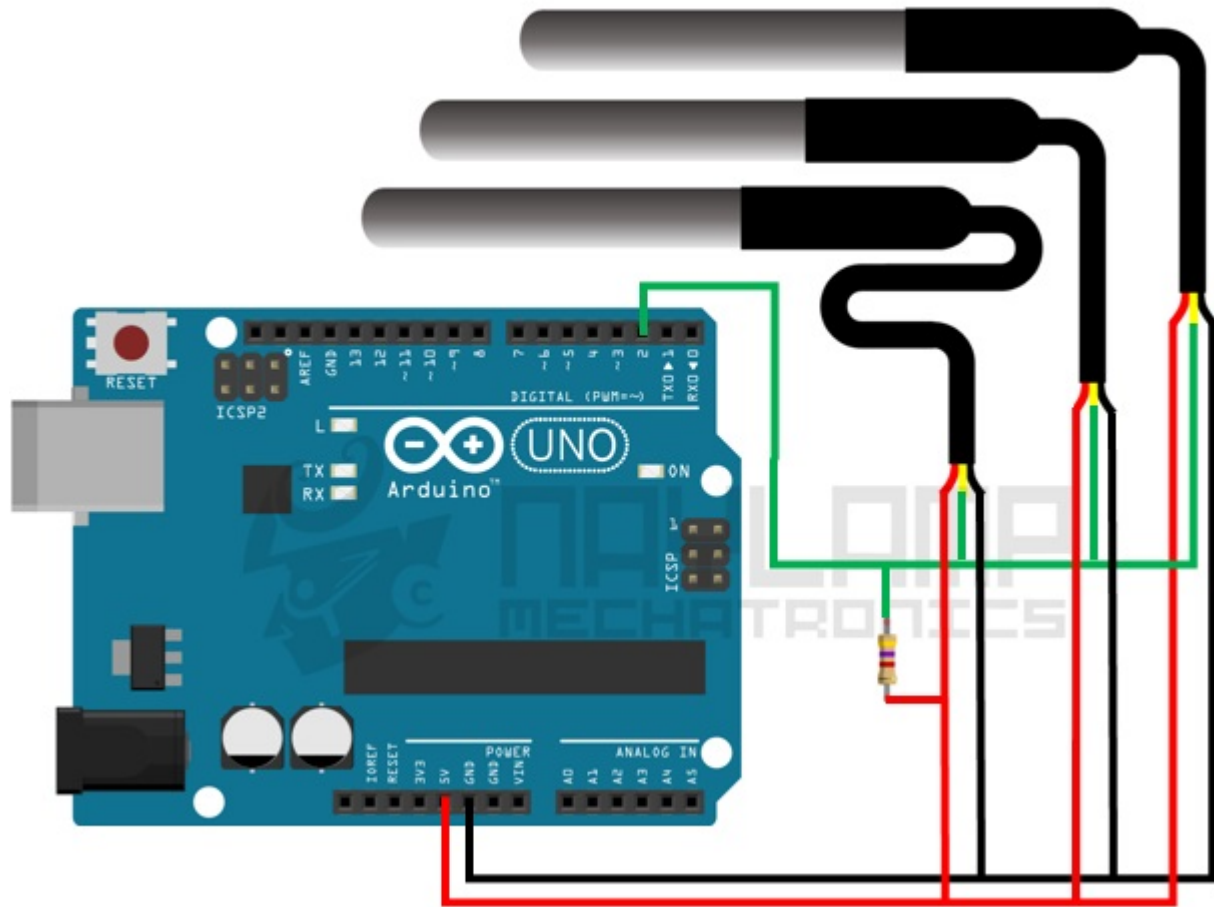
Serial.print("Temperatura 1 = ");
Serial.print(temp1);
Serial.print(" C");
Serial.print("    Temperatura 2 = ");
Serial.print(temp2);
Serial.println(" C");
delay(100);
}
```



Esta forma de conectar dos o más sensores es fácil de entender e implementar y es útil cuando son pocos sensores o simplemente tenemos pines disponibles para conectar más DS18B20 como en un Arduino Mega.

Ej.3: Usando varios DS18B20 con un solo pin del Arduino:

En este caso conectamos todos los sensores al mismo bus 1-Wire.



En caso sea necesario es posible conectar más sensores al mismo pin de datos.

La diferencia acá, es que al tratarse de un bus, necesitamos averiguar la dirección de cada sensor para poder identificarlo.

El siguiente sketch solo se utiliza para obtener la dirección de los dispositivos conectados en el bus 1-wire:

```
#include <OneWire.h>

OneWire ourWire(2);           //Se establece el pin 2 como bus OneWire

void setup(void) {
  Serial.begin(9600);
}

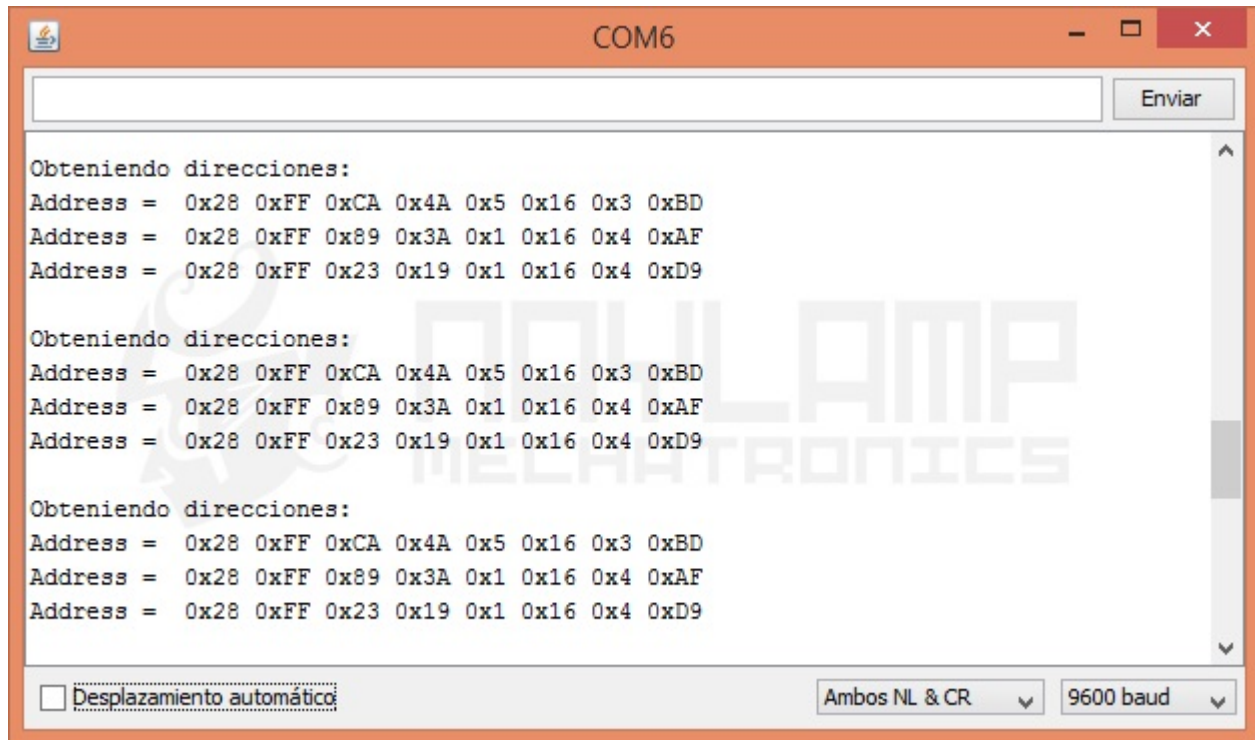
void loop(void) {
  byte addr[8];
  Serial.println("Obteniendo direcciones:");
  while (ourWire.search(addr))
  {
    Serial.print("Address = ");
    for( int i = 0; i < 8; i++) {
      Serial.print(" 0x");
      Serial.print(addr[i], HEX);
    }
  }
}
```

```

    }
    Serial.println();
}

Serial.println();
ourWire.reset_search();
delay(2000);
}

```



El código anterior nos sirve para obtener las direcciones de los sensores, en nuestro caso hemos obtenido las direcciones de los tres sensores que tenemos conectados, pero pueden ejecutar el código anterior para cada sensor de forma individual para saber con exactitud la dirección de su sensor.

Una vez obtenida la dirección podemos identificar la lectura del sensor que deseemos medir.

Para realizar las lecturas usamos el siguiente sketch:

```

#include <OneWire.h>
#include <DallasTemperature.h>

OneWire ourWire(2); //Se establece el pin 2 como bus OneWire

DallasTemperature sensors(&ourWire); //Se declara una variable u objeto para nuestro sensor

DeviceAddress address1 = {0x28, 0xFF, 0xCA, 0x4A, 0x5, 0x16, 0x3, 0xBD}; //dirección del sensor 1
DeviceAddress address2 = {0x28, 0xFF, 0x89, 0x3A, 0x1, 0x16, 0x4, 0xAF}; //dirección del sensor 2
DeviceAddress address3 = {0x28, 0xFF, 0x23, 0x19, 0x1, 0x16, 0x4, 0xD9}; //dirección del sensor 3

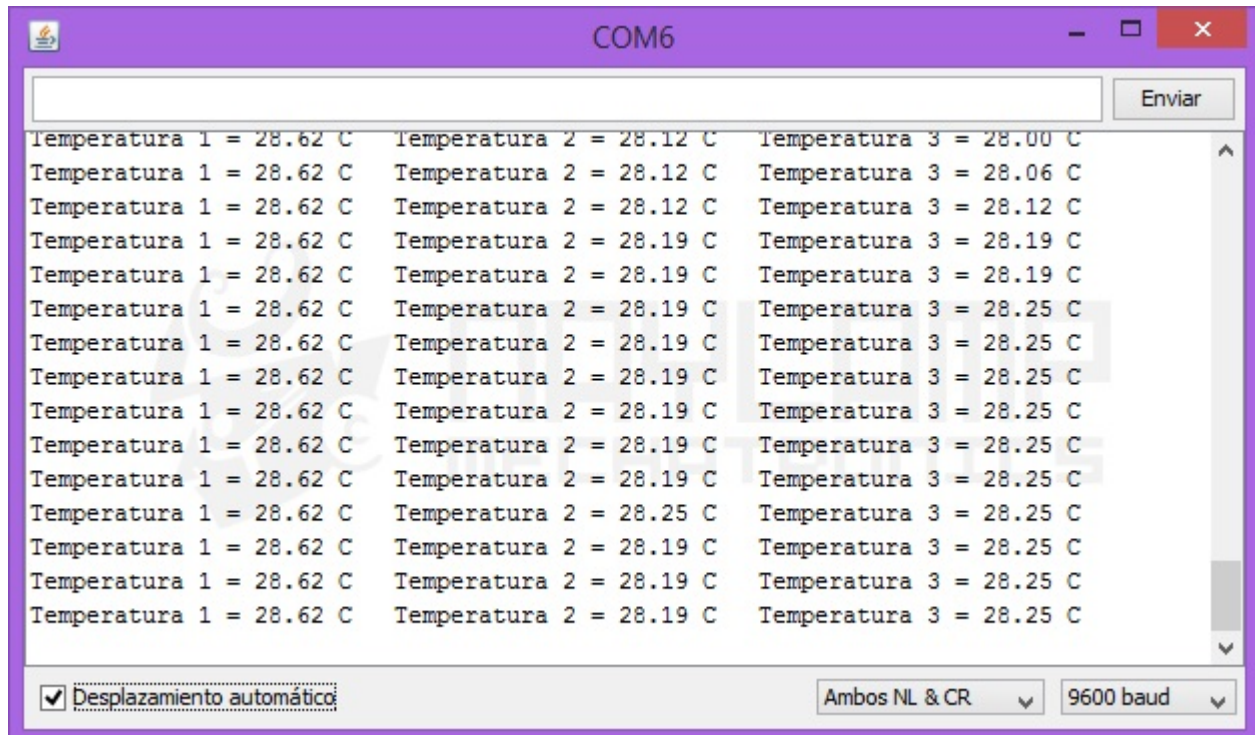
void setup() {
  delay(1000);
}

```

```
Serial.begin(9600);  
sensors.begin(); //Se inicia el sensor  
}  
  
void loop() {  
  
sensors.requestTemperatures(); //envía el comando para obtener las temperaturas  
float temp1= sensors.getTempC(address1); //Se obtiene la temperatura en °C del sensor 1  
float temp2= sensors.getTempC(address2); //Se obtiene la temperatura en °C del sensor 2  
float temp3= sensors.getTempC(address3); //Se obtiene la temperatura en °C del sensor 3  
  
Serial.print("Temperatura 1 = ");  
Serial.print(temp1);  
Serial.print(" C");  
Serial.print("    Temperatura 2 = ");  
Serial.print(temp2);  
Serial.print(" C");  
Serial.print("    Temperatura 3 = ");  
Serial.print(temp3);  
Serial.println(" C");  
delay(100);  
delay(100);  
}
```

Se debe tener en cuenta que se deben reemplazar la direcciones en el sketch con las direcciones correspondientes a los sensores que se tenga disponibles.

A continuación mostramos nuestros resultados:



Las temperaturas sensadas son similares puesto que los sensores se encontraban en el

mismo ambiente:

Para cambiar la resolución del sensor a: 9, 10, 11 o 12 bits. solo se debe usar la función:

```
sensors.setResolution(Address, 9); // resolución de 9 bits
```

Normalmente la resolución se configura en el void setup() después de inicializar los sensores. A menor resolución el tiempo de lectura será menor.

Pueden adquirir los materiales usados en este tutorial en nuestra tienda:

- Arduino Uno R3
- Sensor de Temperatura Digital DS18B20

Tags: Temperatura One Wire

42

?

0

0

0

Compartir

2

14 Comments



eduardo

ago 27, 2016

Excelente tutorial me ha ayudado demasiado, tambien felicitarlo por la pagina y el contenido de esta, una consulta, hasta cuantos sensores es recomendable colocar?

Reply



Naylamp

ago 31, 2016

Hola Eduardo, gracias por el comentario, estaremos publicando más tutoriales. Con respecto a la cantidad de sensores depende de como los conectes. Si usas la conexión del ej.2 puedes conectar los sensores que desea mientras tengas pines suficientes, si conectas en un mismo bus como en el ej.3 puedes trabajar hasta más de 10. Puedes mezclar también las dos formas de conexión, dos o más buses con varios sensores en cada bus.

Reply

**Paulo**

sep 23, 2016

El tutorial que hiciste, es excelente, no me quedo ninguna duda, realmente completo, mostraste todos los tipos de conexiones que necesitaba, te lo agradezco que hicieras el tutorial.

[Reply](#)**Tina**

oct 29, 2016

Hola! muy bueno el tutorial, mi duda es, se puede conectar otro tipo sensor en el bus oneWire, como por ejemplo sensor de PH o sensor de Turbiedad????

[Reply](#)**Naylamp**

nov 5, 2016

Hola Tina, Si se puede y se tiene que trabajar con las direcciones de cada dispositivo, como en el ejemplo 3

[Reply](#)**jonh**

nov 26, 2016

Les recomiendo este software para visualizar los datos que se enian via serial es muy bueno permite enviar hast 9 datos que seran mostrados en en tiempo real. <https://www.youtube.com/watch?v=L1fVNvopPco&t=396s>

[Reply](#)**Eduardo**

ene 18, 2017

Gracias por compartir tu conocimiento con los que iniciamos en arduino, he tenido algunos inconvenientes con las distancias de cables y me han recomendado un mosfet para dar mas potencia al onewire que opinas de esto y cual me recomiendas?y de nuevo muchísimas gracias

[Reply](#)**Naylamp**

feb 1, 2017

Hola Eduardo, con que longitud de cable estas trabajando? el MOSFET es necesario para cundo alimentas el DS18B20 de la línea onewire, si estas usando una alimentación externa (VCC) como en nuestro tutorial no es necesario el MOSFET. Puedes intentar bajar el valor dela resistencia pull-up, por ejemplo para una distancia 20 metros usa una resistencia de 2K. Saludos

[Reply](#)

**facundo**

abr 8, 2017

Hola una consulta se puede utilizar los pines analogicos?

[Reply](#)**Naylamp**

abr 12, 2017

Hola Facundo, si puedes usar los pines analógicos, pues estos se pueden usar también como pines digitales.

[Reply](#)**JuanCubillos**

jun 7, 2017

Hola Naylamp. Una consulta. quiero medir la temperatura de motores trifasico de 50 hp en la carcasa colocando tres de estos en diferentes puntos del motor. El sensor DS18B20 es inmune al ruido electromagnético para estos motores. gracias

[Reply](#)**Nelson Jesus**

jul 30, 2017

hola que tal es muy buena informacion, y abria una forma de almacenar datos en una sd para poder exportarlos a excel

[Reply](#)**Naylamp**

ago 22, 2017

Hola Nelson, en nuestro Tutorial SD explicamos cómo exportar datos de una SD a Excel.

[Reply](#)**Andres Melguizo Cortijo**

sep 5, 2017

Gracias por el tutorial. Me ha sido de mucha ayuda

[Reply](#)**Leave a Reply***** Name:**