

Sensor de temperatura DS18B20 – 18S20P con Arduino (versión 09-7-19)

El sensor de temperatura **DS18B20** es un dispositivo que se comunica de forma digital. Cuenta con tres terminales: Vcc, GND y el pin Data. Este sensor utiliza comunicación por protocolo serial digital OneWire. Este protocolo de comunicación permite enviar y recibir datos utilizando un solo cable. A diferencia de otros, que utilizan dos o más líneas de comunicación digital. Para leer el sensor con un Arduino es necesario utilizar dos librerías que deben ser instaladas antes de cargar el código a nuestra placa de desarrollo. Entonces las librerías son las siguientes:

- Dallas Temperature.
- OneWire

CARACTERÍSTICAS

- Sensor Digital.
- Resolución de 9 y 12 bits.
- Rango de operación de -50 a 125 grados Centígrados.
- Precisión +- 0.5 grados.
- Protocolo OneWire.

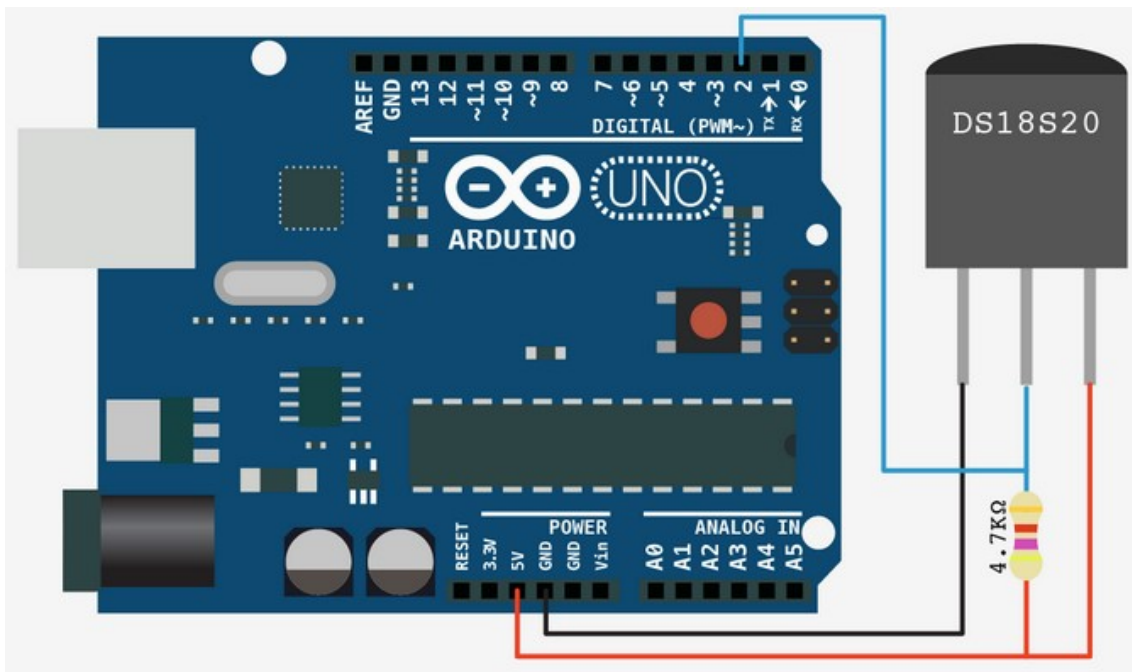
MATERIAL A UTILIZAR

- Placa Arduino UNO.
- Cables Jumper.
- Sensor DS18B20.
- Protoboard.
- Resistencia 4.7 K.

DIAGRAMA DE CONEXIONES

El diagrama de conexiones para el DS18B20 y su correcto funcionamiento del sensor, hay que poner una resistencia de 4.7K del pin de Datos y Vcc. Por ejemplo, normalmente este sensor viene blindado. Además consta de un cable largo para aplicaciones donde es necesario sumergirlo en líquidos u otras sustancias. Entonces, esta presentación del sensor sólo trae 3 terminales o cables de conexión. Finalmente el pin de Vcc es el cable Rojo, GND es el cable Negro y el Cable de datos puede ser de color Amarillo o Blanco.

Figura 1. Esta imagen muestra la conexión del Arduino con el sensor DS18B20



CÓDIGO DE EJEMPLO

```
#include <OneWire.h> //Se importan las librerías
#include <DallasTemperature.h>
#define Pin 2 //Se declara el pin donde se conectará la
DATA
OneWire ourWire(Pin); //Se establece el pin declarado como bus
para la comunicación OneWire
DallasTemperature sensors(&ourWire); //Se llama a la librería DallasTemperature

void setup() {
  delay(1000);
  Serial.begin(9600);
  sensors.begin(); //Se inician los sensores
}

void loop() {
  sensors.requestTemperatures(); //Prepara el sensor para la lectura

  Serial.print(sensors.getTempCByIndex(0)); //Se lee e imprime la temperatura en
  grados Centigrados
  Serial.println(" Grados Centigrados");

  delay(1000); //Se provoca una parada de 1 segundo antes
  de la próxima lectura
}
```

RESULTADOS EN MONITOR SERIAL DE ARDUINO (IMAGEN DE INTERNET)

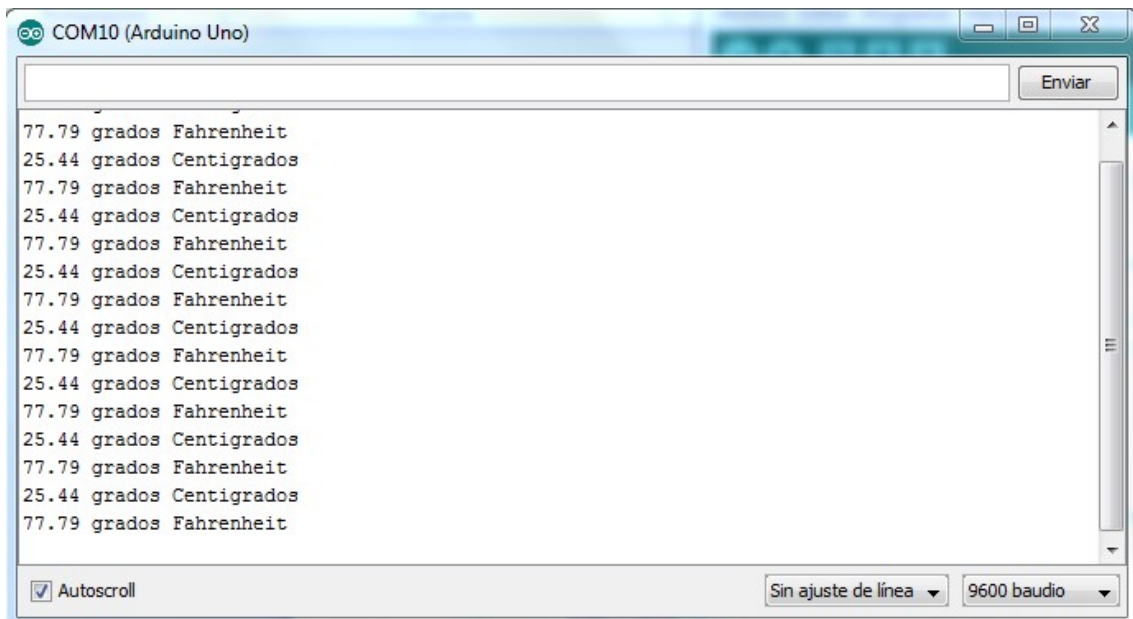


Figura 2. Aquí se muestra la temperatura en grados fahrenheit y grados centigrados.

Enlaces de Interés

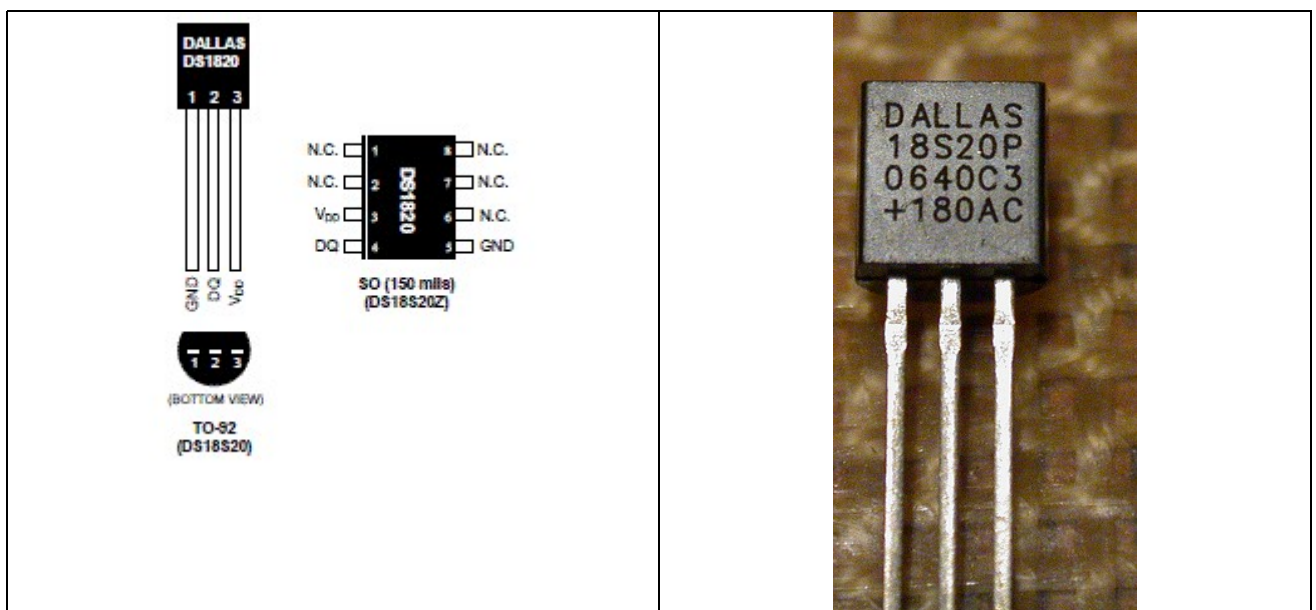
HOJA DE DATOS DEL SENSOR

<http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>

LIBRERÍAS DALLAS TEMPERATURA Y ONEWIRE

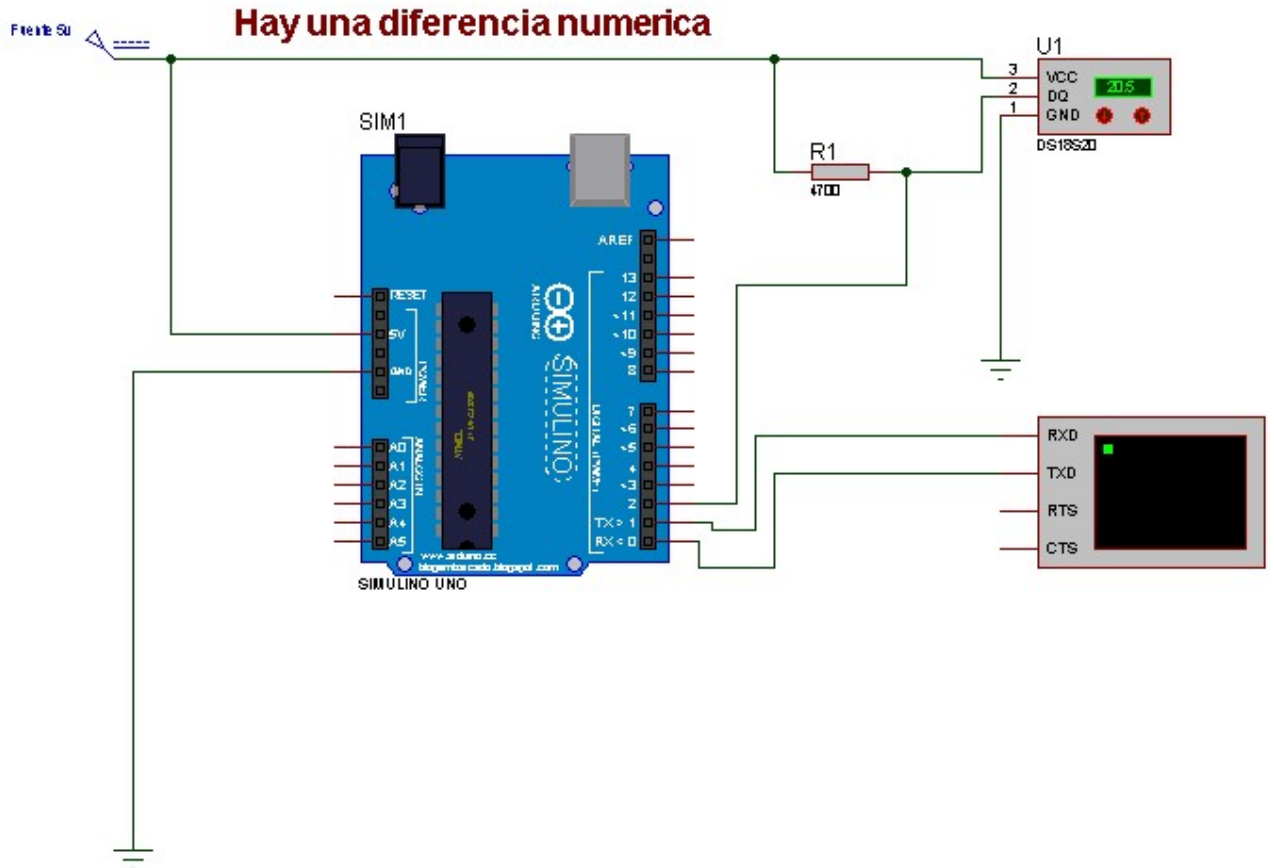
<https://github.com/milesburton/Arduino-Temperature-Control-Library>

<https://github.com/ntruchsess/arduino-OneWire>



Ensayos en Proteus

DS18S20 - Adaptado a Proteus (ESPERE PARA MOSTRAR DATOS)



Programa de prueba en Proteus

```
//Variante Prof: Bolaños DJ version 9-7-19 Para PROTEUS
/*
 * Se presenta un problema al simular en Proteus
 * Obtengo algunos valores correctos, pero la mayoría de los datos
 * es solo el valor de -127.00
 * No habiendo encontrado una solución distinta al día de hoy
 * se agrega un IF que contrala ese dato erroneo en Proteus y se excluye.
 */

#include <OneWire.h>           //Se importan las librerías
#include <DallasTemperature.h>
#define Pin 2                 //Se declara el pin donde se conectará la DATA
OneWire ourWire(Pin);        //Se establece el pin declarado como bus para la comunicación
OneWire
DallasTemperature sensors(&ourWire); //Se llama a la librería DallasTemperature
float temper;
void setup() {

delay(1000);
Serial.begin(9600);
sensors.begin();           //Se inician los sensores
}

void loop() {
sensors.requestTemperatures(); //Prepara el sensor para la lectura
```

```
temper =sensors.getTempCByIndex(0);//Se lee la temperatura

if (temper > -127) //Probado en Proteus
{

    Serial.print(temper); //Se imprime la temperatura en grados Centigrados
    Serial.println(" Grados Centigrados");

}

delay(1000); // Se provoca una parada de 1 segundo (original) antes de la próxima lectura

}
```