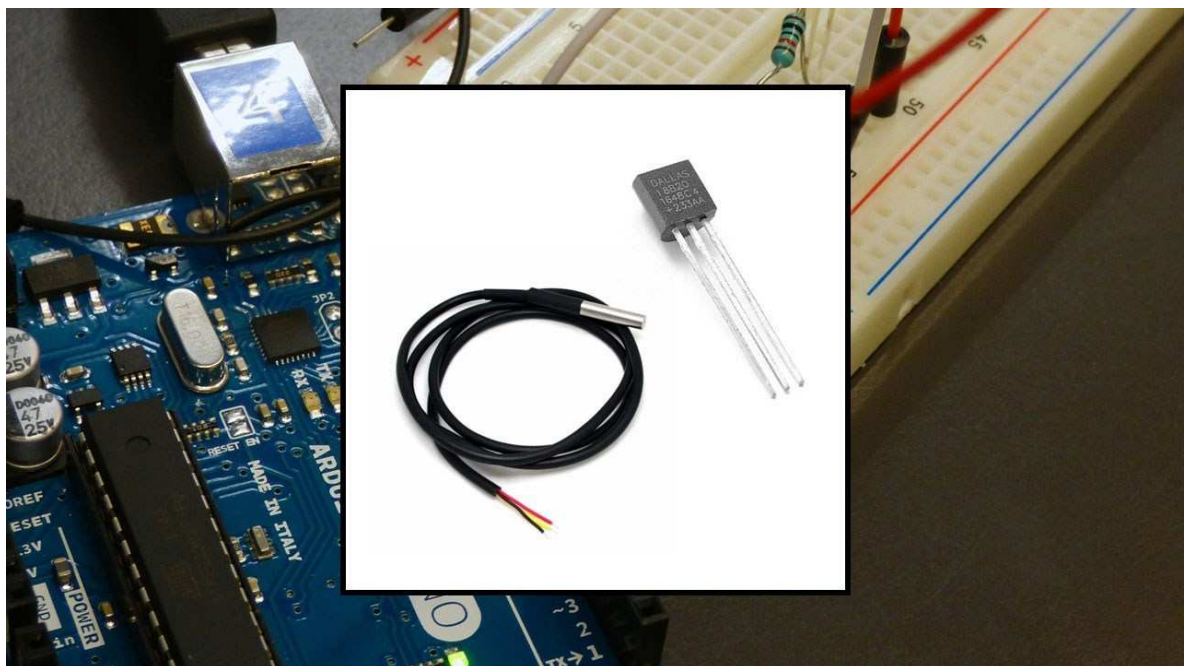


Guide for DS18B20 Temperature Sensor with Arduino

This guide shows how to use the DS18B20 temperature sensor with the Arduino board. You'll learn how to wire the sensor, install the required libraries and get temperature from one or multiple DS18B20 sensors.



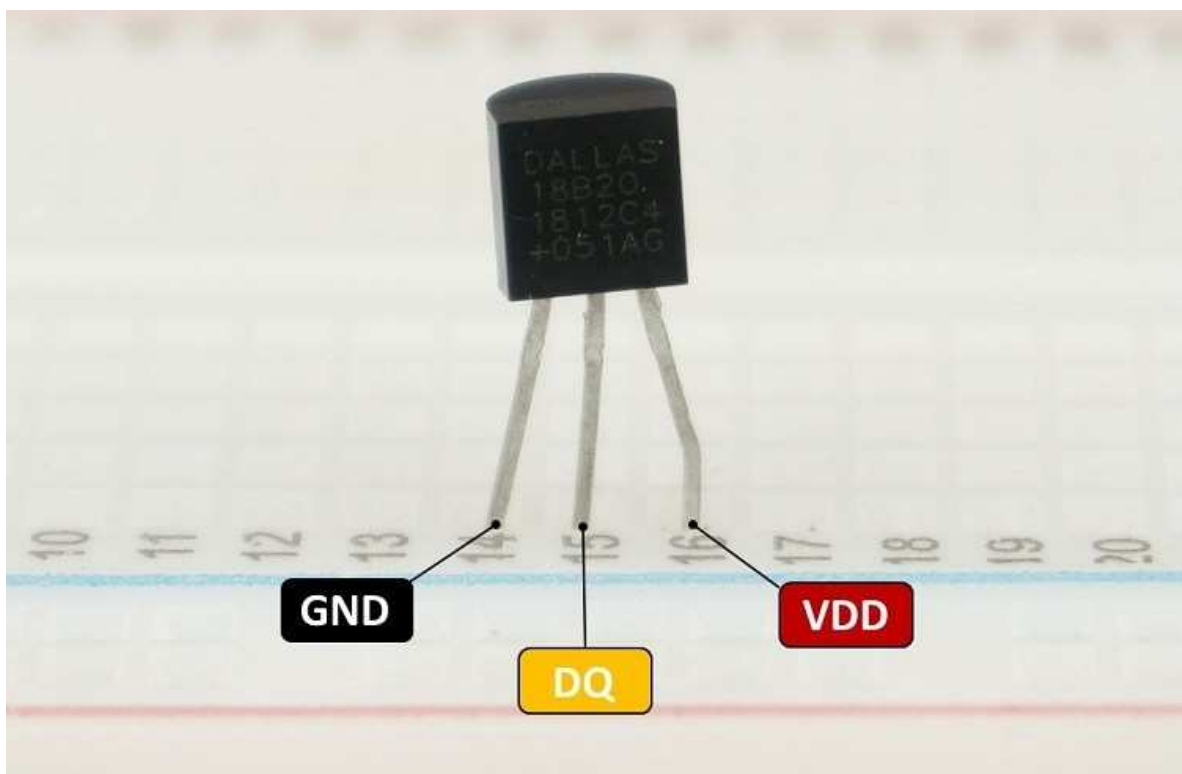
You might also like reading other DS18B20 guides:

- [ESP32 DS18B20 Temperature Sensor with Arduino IDE](#)
- [ESP8266 DS18B20 Temperature Sensor with Arduino IDE](#)
- [ESP32/ESP8266 DS18B20 Temperature Sensor with MicroPython](#)
- [ESP32 with Multiple DS18B20 Temperature Sensors](#)

Introducing DS18B20 Temperature Sensor

The [DS18B20 temperature sensor](#) is a one-wire digital temperature sensor. This means that it just requires one data line (and GND) to communicate with the Arduino.

It can be powered by an external power supply or it can derive power from the data line (called “parasite mode”), which eliminates the need for an external power supply.



The following table shows how you should wire the DS18B20 sensor to your Arduino board:

DS18B20	Arduino
GND	GND
DQ	Any digital pin (with 4.7k Ohm pull-up resistor)
VDD	5V (normal mode) or GND (parasite mode)

Each DS18B20 temperature sensor has a unique 64-bit serial code. This allows you to wire multiple sensors to the same data wire. So, you can get temperature from multiple sensors using just one Arduino digital pin.

The DS18B20 temperature sensor is also available in [waterproof version](#).



Here's a summary of the most relevant specs of the DS18B20 temperature sensor:

- Communicates over one-wire bus communication
- Power supply range: 3.0V to 5.5V
- Operating temperature range: -55°C to +125°C
- Accuracy +/-0.5 °C (between the range -10°C to 85°C)

For more information consult the [DS18B20 datasheet](#).

Parts Required

To show you how the sensor works, we'll build a simple example that reads the temperature from the DS18B20 sensor with the Arduino and displays the values on the Arduino Serial Monitor.

Here's a list of parts you need to complete this tutorial

- [Arduino UNO](#) – read [Best Arduino Starter Kits](#)
- [DS18B20 temperature sensor](#) (one or multiple sensors) – [waterproof version](#)
- [4.7k Ohm resistor](#)
- [Breadboard](#)
- [Jumper wires](#)

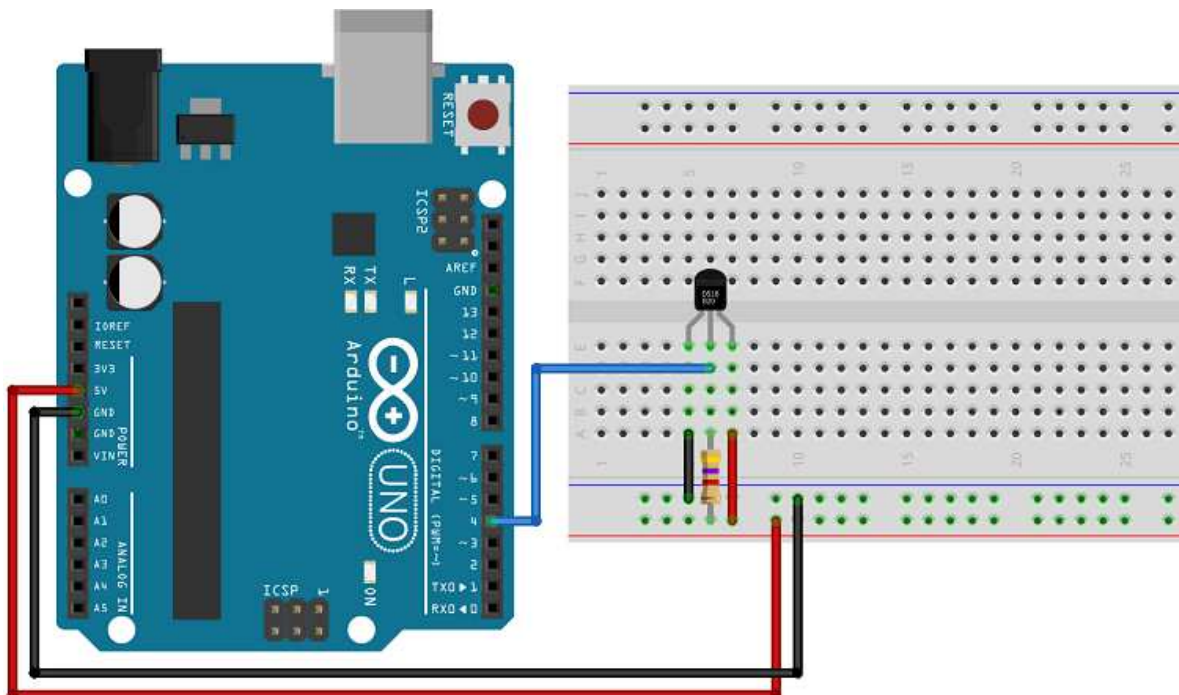
You can use the preceding links or go directly to [MakerAdvisor.com/tools](https://makeradvisor.com/tools) to find all the parts for your projects at the best price!



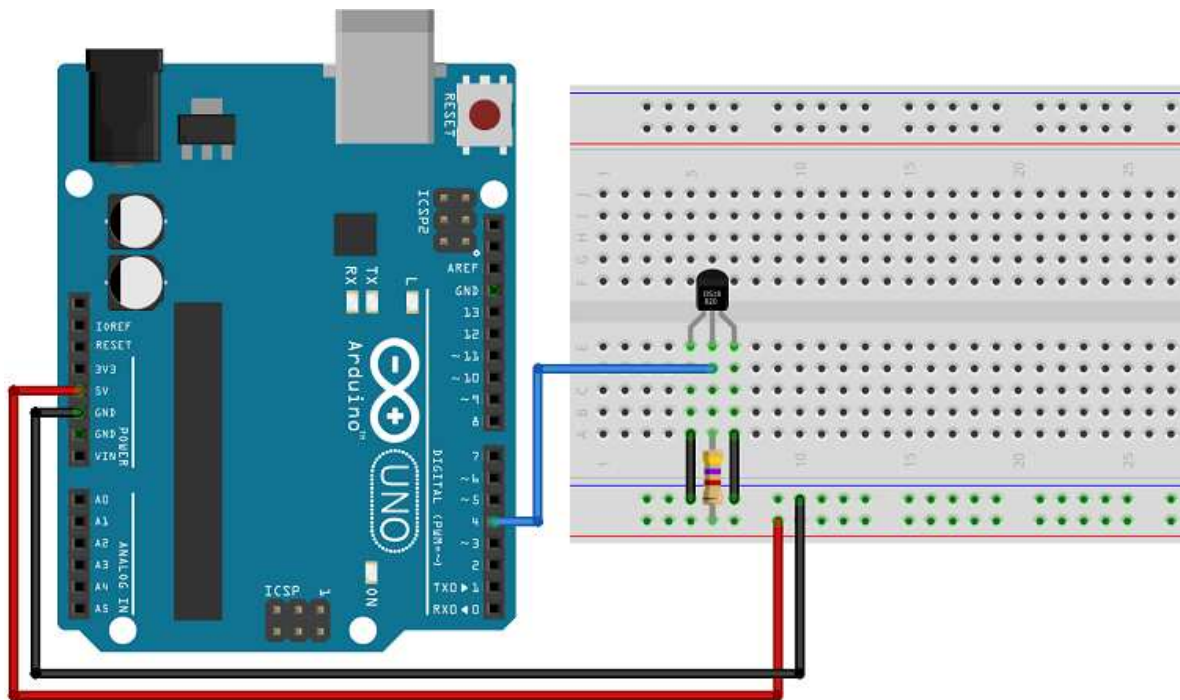
Schematic

The sensor can operate in two modes:

- **Normal mode:** 3-wire connection is needed. You provide power to the VDD pin. Here's the schematic you need to follow:



- **Parasite mode:** You only need data and GND. The sensor derives its power from the data line. In this case, here's the schematic you need to follow:



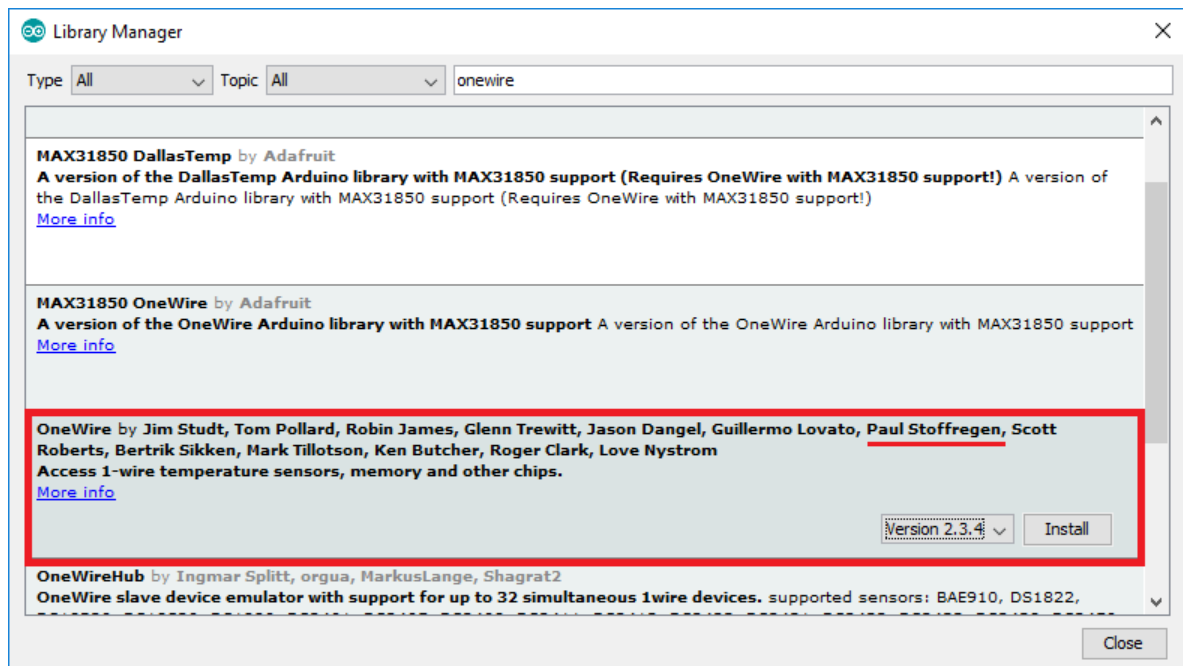
You can read the temperature of more than one sensor at the same time using just one Arduino digital pin. For that, you just need to wire together all the sensors data pins to an Arduino digital pin.

Upload Code – Single DS18B20

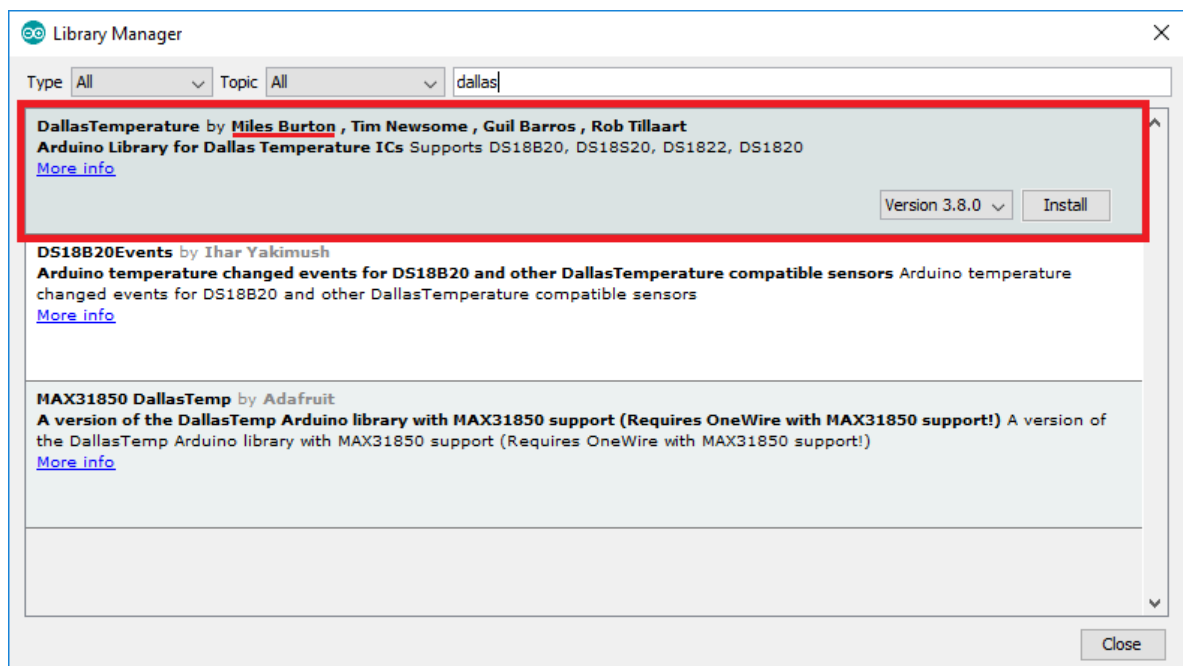
To interface with the DS18B20 temperature sensor, you need to install the [One Wire library by Paul Stoffregen](#) and the [Dallas Temperature library](#). Follow the next steps to install those libraries.

Installing Libraries

1. Open your Arduino IDE and go to **Sketch > Include Library > Manage Libraries**. The Library Manager should open.
2. Type “**OneWire**” in the search box and install the OneWire library by Paul Stoffregen.



3. Then, search for “**Dallas**” and install the Dallas Temperature library by Miles Burton.



After installing the needed libraries, upload the following code to your Arduino board. This sketch is based on an example from the Dallas Temperature library.

```
/******
```

Rui Santos

Complete project details at <https://randomnerdtutorials.com>

Based on the Dallas Temperature Library example

*****/

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

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

```
// Data wire is conntec to the Arduino digital pin 4
```

```
#define ONE_WIRE_BUS 4
```

```
// Setup a oneWire instance to communicate with any OneWire devices
```

```
OneWire oneWire(ONE_WIRE_BUS);
```

```
// Pass our oneWire reference to Dallas Temperature sensor
```

```
DallasTemperature sensors(&oneWire);
```

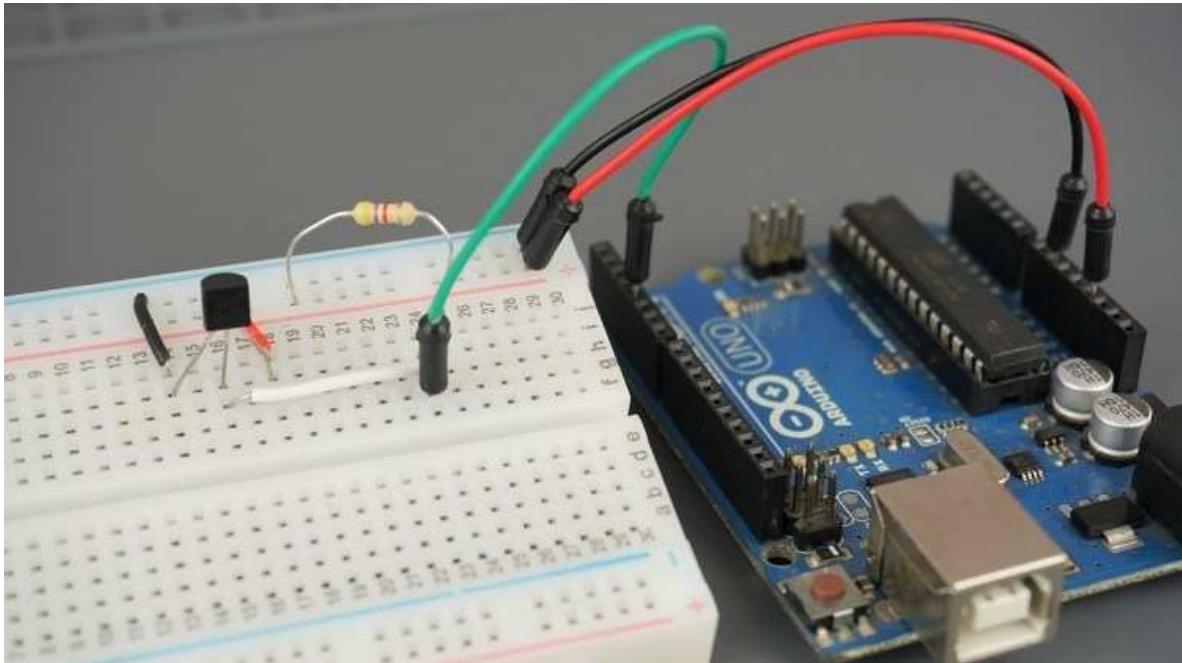
```
void setup(void)
```

```
{
```

```
    // Start serial communication for debugging purposes
```

[View raw code](#)

There are many different ways to get the temperature from DS18B20 temperature sensors. If you're using just one single sensor, this is one of the easiest and simplest ways.



How the Code Works

Start by including the `OneWire` and the `DallasTemperature` libraries.

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

Create the instances needed for the temperature sensor. The temperature sensor is connected to Pin 4.

```
// Data wire is conntec to the Arduino digital pin 4
const int oneWireBus = 4;
// Setup a oneWire instance to communicate with any OneWire devices
OneWire oneWire(oneWireBus);
// Pass our oneWire reference to Dallas Temperature sensor
DallasTemperature sensors(&oneWire);
```

In the `setup()` , initialize the Serial Monitor at a baud rate of 9600.

```
Serial.begin(9600);
```


Initialize the DS18B20 temperature sensor:

```
sensors.begin();
```

In the `loop()` is where you'll get the temperature. You need to call the `requestTemperatures()` method before getting the actual temperature value.

```
sensors.requestTemperatures();
```

Then, get and print the temperature in Celsius. To get the temperature in Celsius, use the `getTempCByIndex()` method :

```
Serial.print(sensors.getTempCByIndex(0));
```

Or use the `getTempFByIndex()` to get the temperature in Fahrenheit.

```
Serial.println(sensors.getTempFByIndex(0));
```

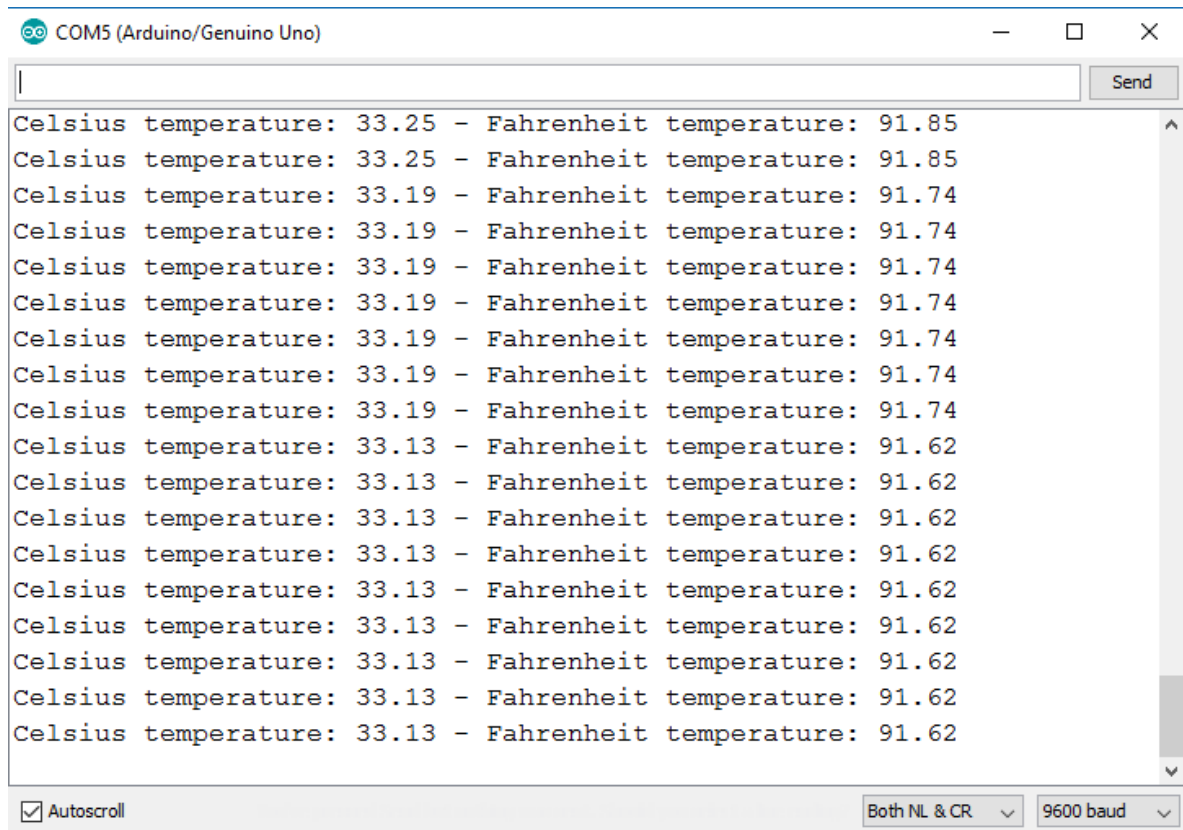
The `getTempCByIndex()` and the `getTempFByIndex()` methods accept the index of the temperature sensor. Because we're using just one sensor its index is 0. If you have more than one sensor, you use index 0 for the first sensor, index 1 for the second sensor, and so on.

New temperature readings are requested every second.

```
delay(5000);
```

Demonstration

After uploading the code, open the Arduino IDE Serial Monitor at a 9600 baud rate. You should get the temperature displayed in both Celsius and Fahrenheit:

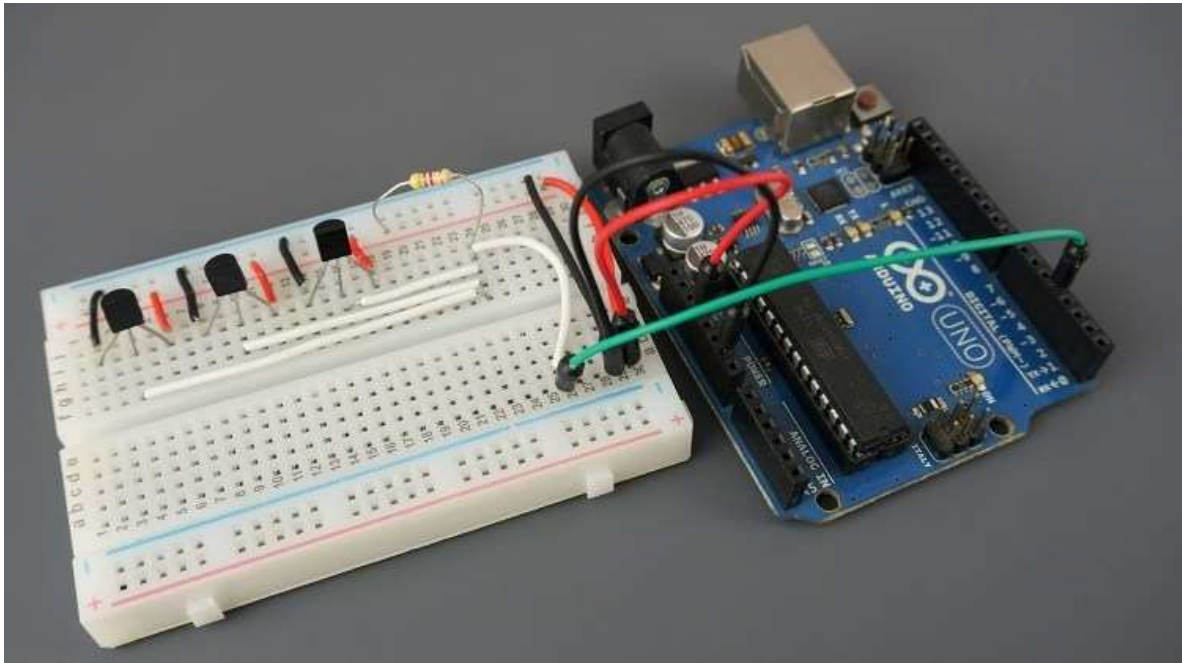


The screenshot shows the Arduino IDE Serial Monitor window for a COM5 (Arduino/Genuino Uno) connection. The window displays a series of temperature readings in both Celsius and Fahrenheit. The data is as follows:

Celsius temperature	Fahrenheit temperature
33.25	91.85
33.25	91.85
33.19	91.74
33.19	91.74
33.19	91.74
33.19	91.74
33.19	91.74
33.19	91.74
33.19	91.74
33.13	91.62
33.13	91.62
33.13	91.62
33.13	91.62
33.13	91.62
33.13	91.62
33.13	91.62
33.13	91.62
33.13	91.62

The window also features a 'Send' button, an 'Autoscroll' checkbox (checked), and dropdown menus for 'Both NL & CR' and '9600 baud'.

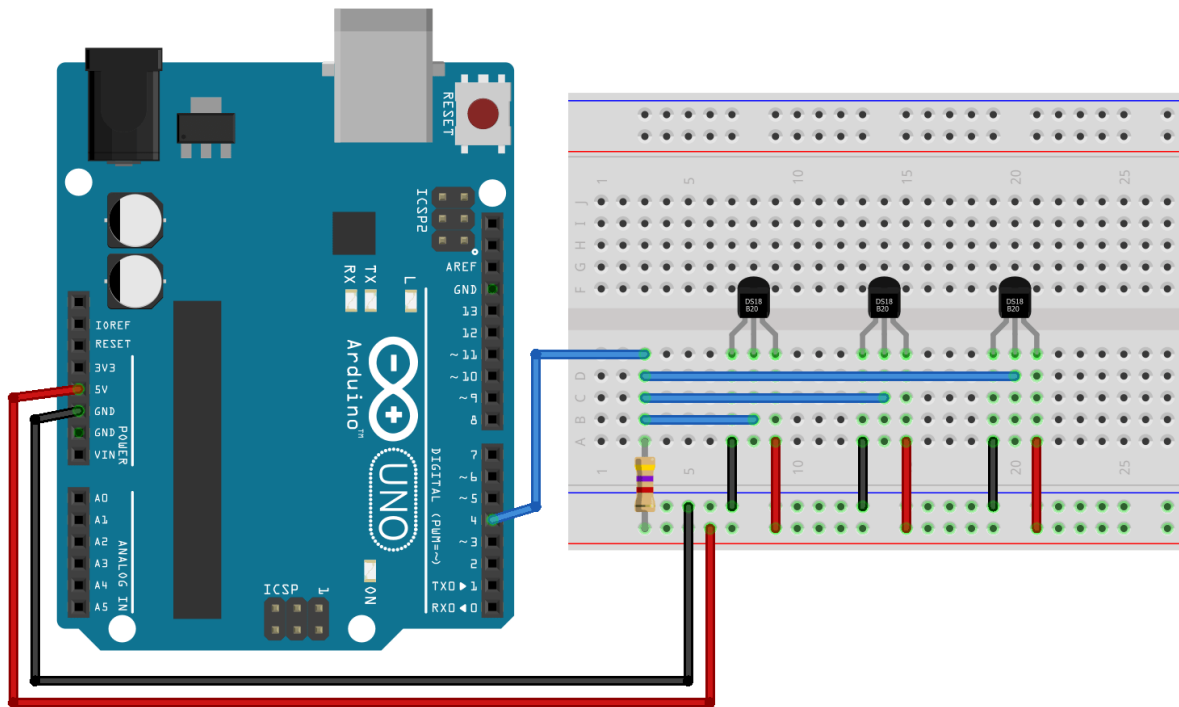
Getting Temperature from Multiple DS18B20 Sensors



The DS18B20 temperature sensor communicates using one-wire protocol and each sensor has a unique 64-bit serial code, so you can read the temperature from multiple sensors using just one single Arduino digital Pin.

Schematic

To read the temperature from multiple sensors, you just need to wire all data lines together as shown in the following schematic diagram:



Upload Code – Multiple DS18B20

Then, upload the following code. It scans for all devices on Pin 4 and prints the temperature for each one. This sketch is based on the example provided by the DallasTemperature library.

```
/*
 * Rui Santos
 * Complete Project Details https://randomnerdtutorials.com
 */

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

// Data wire is plugged into port 4 on the Arduino
#define ONE_WIRE_BUS 4

// Setup a oneWire instance to communicate with any OneWire devices
OneWire oneWire(ONE_WIRE_BUS);

// Pass our oneWire reference to Dallas Temperature.
```

[View raw code](#)

How the code works

The code uses several useful methods to handle multiple DS18B20 sensors.

You use the `getDeviceCount()` method to get the number of DS18B20 sensors on the data line.

```
numberOfDevices = sensors.getDeviceCount();
```

The `getAddress()` method finds the sensors addresses:

```
if(sensors.getAddress(tempDeviceAddress, i)){
```

The address is unique for each sensor. So each sensor can be identified by its address.

Then, you use the `getTempC()` method that accepts as argument the device address. With this method you can get the temperature from a specific sensor:

```
float tempC = sensors.getTempC(tempDeviceAddress);
```

To get the temperature in Fahrenheit degrees, you can use the `getTempF()`. Alternatively, you can convert the temperature in Celsius to Fahrenheit as follows:

```
DallasTemperature::toFahrenheit(tempC)
```

Wrapping Up

The DS18B20 temperature sensor is a one-wire digital sensor. To use this sensor with the Arduino, you need the OneWire and the DallasTemperature libraries. You can use one sensor or multiple sensors on the same data line because you can identify each sensor by its unique address.

Now, you can take this project further and display your [sensor readings in an OLED display](#), for example.

We have more tutorials for other Arduino compatible sensors that you may find useful:

- [DHT11/DHT22 Humidity and Temperature Sensor With Arduino](#)
- [BME280 \(Temperature, Humidity, Pressure\) with Arduino](#)
- [Relay Module with Arduino](#)
- [Ultrasonic Sensor HC-SR04 with Arduino](#)

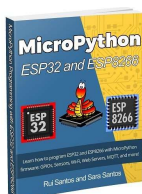
We hope you've found this guide useful.

If you want to learn more about Arduino, take a look at our resources:

- [Arduino Step-by-step projects course](#)
- [Free Arduino Projects and Tutorials](#)
- [Arduino Mini Course](#)

Thanks for reading!

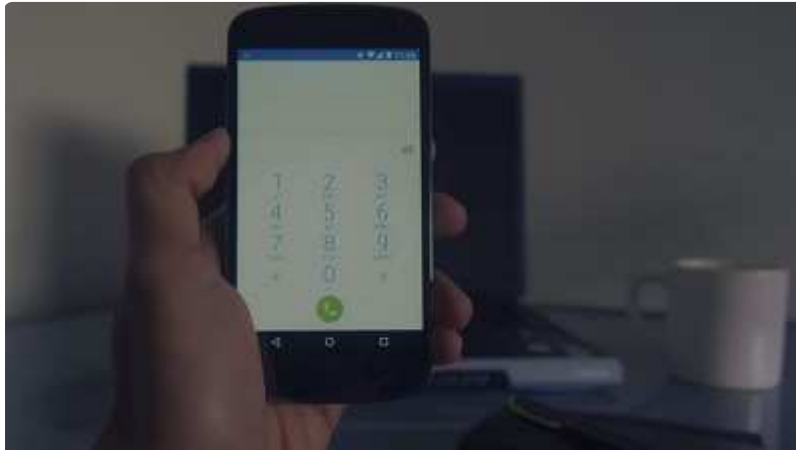
Updated July 2, 2019



[eBook] MicroPython Programming with ESP32 and ESP8266

Learn how to program and build projects with the ESP32 and ESP8266 using MicroPython firmware [DOWNLOAD »](#)

Recommended Resources



[Build a Home Automation System from Scratch »](#) With Raspberry Pi, ESP8266, Arduino, and Node-RED.



[Home Automation using ESP8266 eBook and video course »](#) Build IoT and home automation projects.



[Arduino Step by Step Projects »](#) Build 25 Arduino projects with

What to Read Next...



ESP8266 Remote Controlled Sockets



ESP8266 NodeMCU Access Point (AP) for Web Server

ESP32-CAM Troubleshooting Guide: Most Common Problems Fixed



Getting Started with ESP-NOW (ESP8266 NodeMCU with Arduino IDE)

ESP32 DHT11/DHT22 Web Server – Temperature and Humidity using Arduino IDE



ESP8266 Web Server with Arduino IDE

0.5

Enjoyed this project? Stay updated by subscribing our weekly newsletter!

✉ SUBSCRIBE

22 thoughts on “Guide for DS18B20 Temperature Sensor with Arduino”



chevalier pierre

August 19, 2016 at 10:23 am

Hello J uses your tutorial . J have errors at compile on dallastemperature.h Can you help me by giving me the dallastemperature library works

Best regards

Pierre

[Reply](#)



Rui Santos

August 24, 2016 at 8:57 am

What's the error?

[Reply](#)



pierre chevalier

August 28, 2016 at 7:44 pm

hello,
error is

error: WConstants.h: No such file or directory

why

do you speak and write french

best regards

[Reply](#)



Rodrigo Catarino

August 20, 2018 at 4:59 pm

Hi Rui,

Thanks for your tutorial.

I need to compare two temperature sensors, lets say T1 and T2 and control a relay when T1 e higher than T2.
How can i achieve this?

Kind regards,
Rodrigo Catarino

[Reply](#)



Sara Santos

August 21, 2018 at 8:50 am

Hi Rodrigo. You need an if statement that compares both temperatures and turns the relay on when $T1 > T2$.
Controlling a relay is the same thing as controlling an LED but with inverted logic.
`digitalWrite(relay, HIGH);` -> turns the relay off
`digitalWrite(relay, LOW);` -> turns the relay on

You may want to take a look at this tutorial to see how to wire the relay to the Arduino:

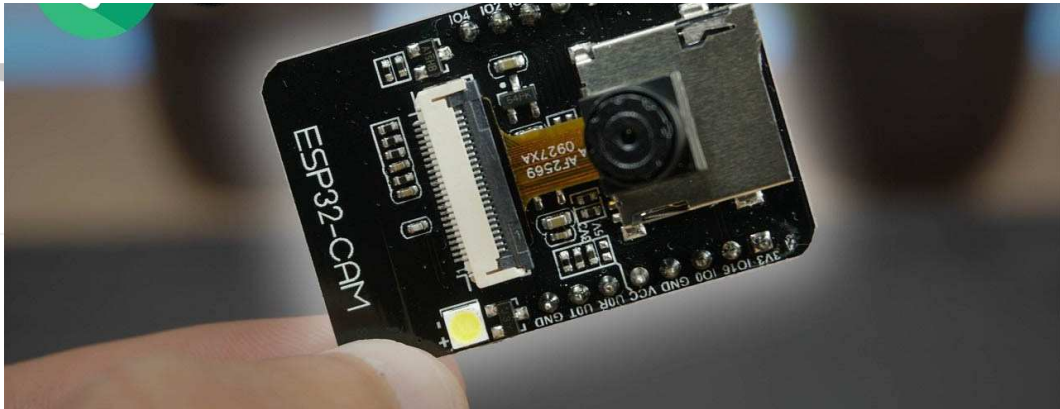
<https://randomnerdtutorials.com/guide-for-relay-module-with-arduino/>

I hope this helps.

Regards,

Sara 😊

[Reply](#)



variation.

[Reply](#)



Regards,
Sara

[Reply](#)



Guneau François

June 18, 2019 at 2:38 am

Hello,
I'm working on a project using an Arduino Zero that as an

Operating Voltage at 3.3 V. Do you think it may varied the pull-up resistor value?
Thanks you !

[Reply](#)



Sara Santos

June 18, 2019 at 8:46 am

Hi.
I think you can use the same resistor and it will work fine.
Regards,
Sara

[Reply](#)



Ademar Lino Kleinubing

July 26, 2019 at 4:54 pm

EXCELENTE POSTAGEM, PARABÉNS.

[Reply](#)



Sara Santos

July 27, 2019 at 11:31 am

Thank you 😊

[Reply](#)



Mielp

July 27, 2019 at 12:05 pm

Thank you so much, was almost giving up on this one.

[Reply](#)



Adam

December 19, 2019 at 3:04 pm

As of 19/12/2019 this code works
thanks.

[Reply](#)



JOSE GUSTAVO ABREU MURTA

December 20, 2019 at 1:11 pm

Great Tutorial ! Thanks a lot.
DS18B20 have four resolutions of reading (9,10,11 and 12 bits).
You may change the resolution of DS18B20 with this command.

<https://github.com/milesburton/Arduino-Temperature-Control-Library/blob/master/examples/Single/Single.pde>
example (12bits resolution):
`sensors.setResolution(insideThermometer, 12);`

Regards.

[Reply](#)



Alan Wild

January 12, 2020 at 11:24 pm

While compiling your temperature sensor sketch
(<https://randomnerdtutorials.com/guide-for-ds18b20-temperature-sensor-with-arduino/>)
I get this error message:
Sketch uses 5924 bytes (18%) of program storage space.
Maximum is 32256 bytes.
Global variables use 290 bytes (14%) of dynamic memory,
leaving 1758 bytes for local variables. Maximum is 2048 bytes.

[Reply](#)



Sara Santos

January 13, 2020 at 2:00 pm

Hi Alan.
That's a normal message when you compile the code.
Regards,
Sara

[Reply](#)



Alan Wild

January 13, 2020 at 2:16 pm

Ok, but it won't download to the arduino because of that.

[Reply](#)



prathamesh raut

February 7, 2020 at 5:31 pm

where is zip or header file of ds18b20
plz me link of that header file or zip file

[Reply](#)



Sara Santos

February 10, 2020 at 10:24 am

You need this two libraries:

<https://github.com/PaulStoffregen/OneWire>

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

Regards,

Sara

[Reply](#)



Vani Rajan

March 11, 2020 at 11:29 am

Hi, I'm getting the values like -127 C and -196.60 F. what will be the issue? can you please help me?

[Reply](#)



Sara Santos

March 11, 2020 at 8:02 pm

Hi.
There's probably something wrong with your circuit wiring.
Regards,
Sara

[Reply](#)



ajdin

March 15, 2020 at 10:41 am

i have the same problem, if you have found a fix please share. i have tried couple of temperature sensors. but the temp. is still -127. and the circuit wiering is ok.

[Reply](#)

Leave a Comment

Name *

Email *

Website

☐ Notify me of follow-up comments by email.

☐ Notify me of new posts by email.



[Arduino Step-by-Step Projects »](#)

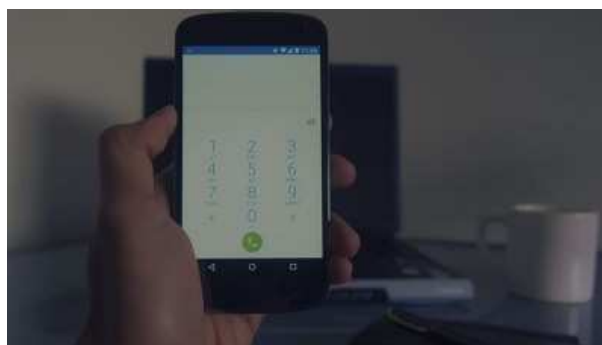
Build 25 cool Arduino projects with our course even with no prior experience!



[Visit Maker Advisor – Tools and Gear for makers, hobbyists and DIYers »](#)



[Home Automation using ESP8266 eBook and video course »](#) Build IoT and home automation projects.



[Build a Home Automation System from Scratch »](#) With Raspberry Pi, ESP8266, Arduino, and Node-RED.

[About](#) [Support](#) [Terms](#) [Privacy](#) [Refunds](#) [MakerAdvisor.com](#) [Join the Lab](#)

Copyright © 2013-2020 · RandomNerdTutorials.com · All Rights Reserved

[Update Privacy Settings](#)

Exclusive Member of Mediavine Home