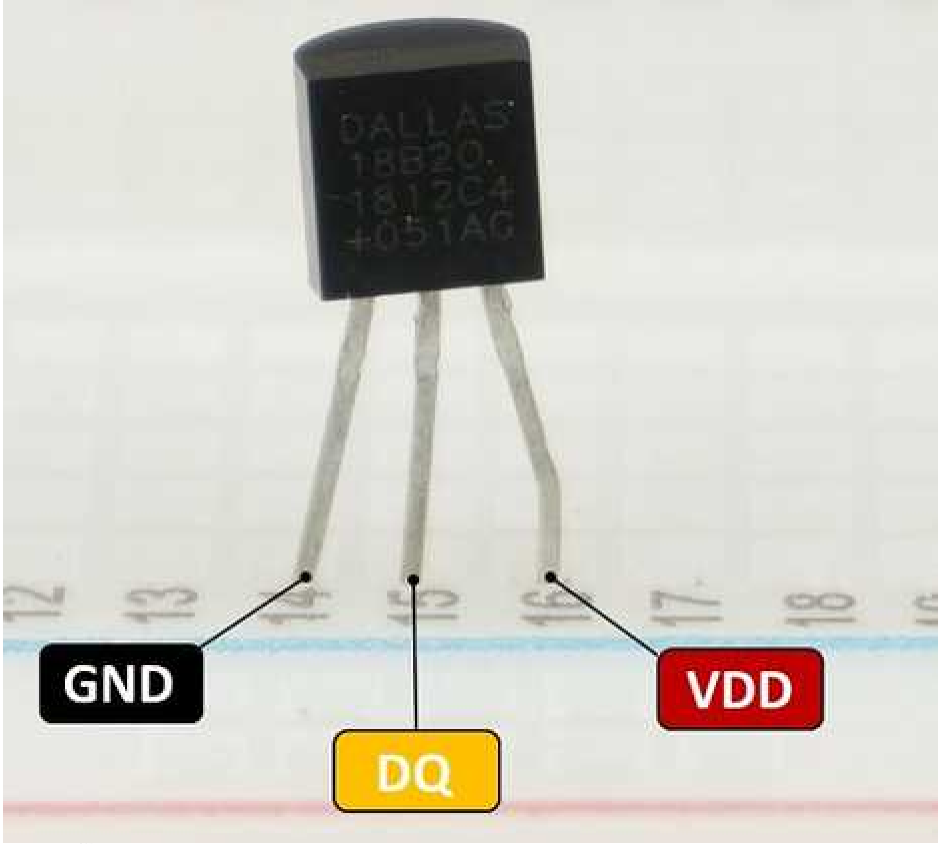
Guide for DS18B20 Temperature Sensor with Arduino

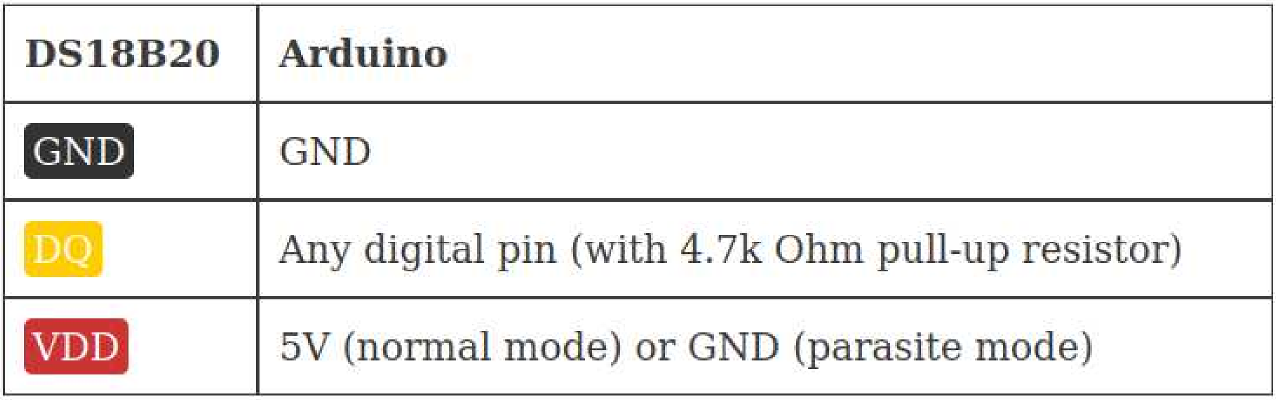
This guide shows how to use the DS18B20 temperature sensor with the Arduino board. You’ll learn how to wire the sensor, install the required libraries and get temperature from one or multiple DS18B20 sensors.

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.





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:

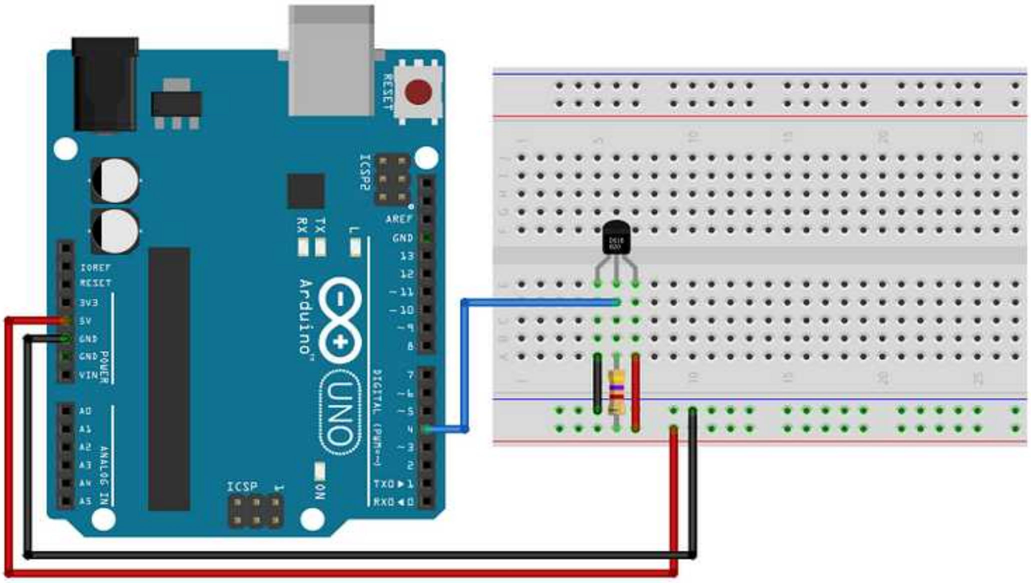
- Communicates over one-wire bus communication
- Power supply range: 3.0V to 5.5V
- Operating temperature range: -55ºC to +125ºC
- Accuracy +/-0.5 ºC (between the range -10ºC to 85ºC)

For more information consult the DS18B20 datasheet.

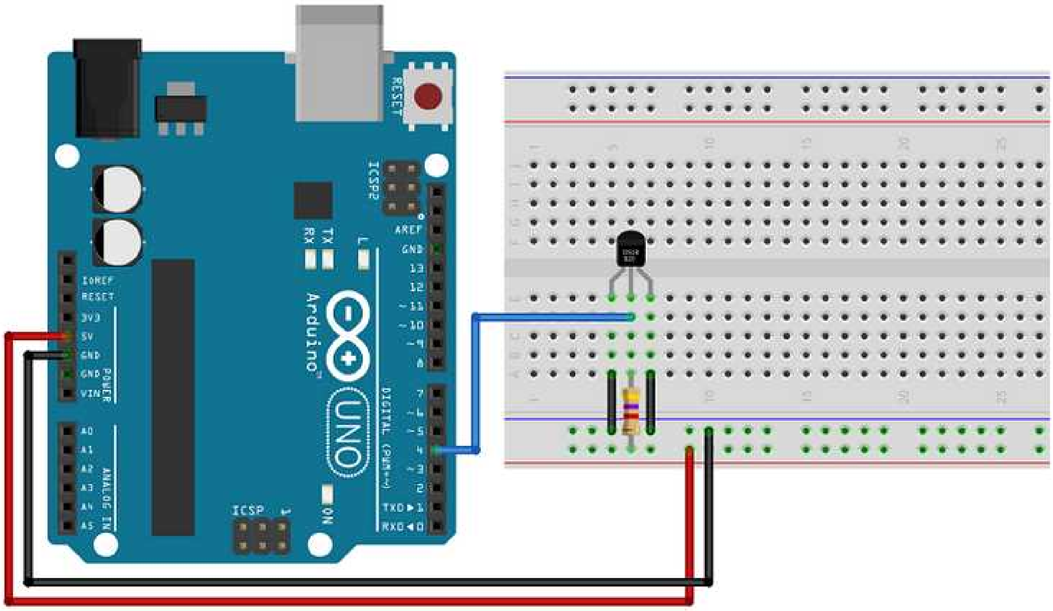
Schematic

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.

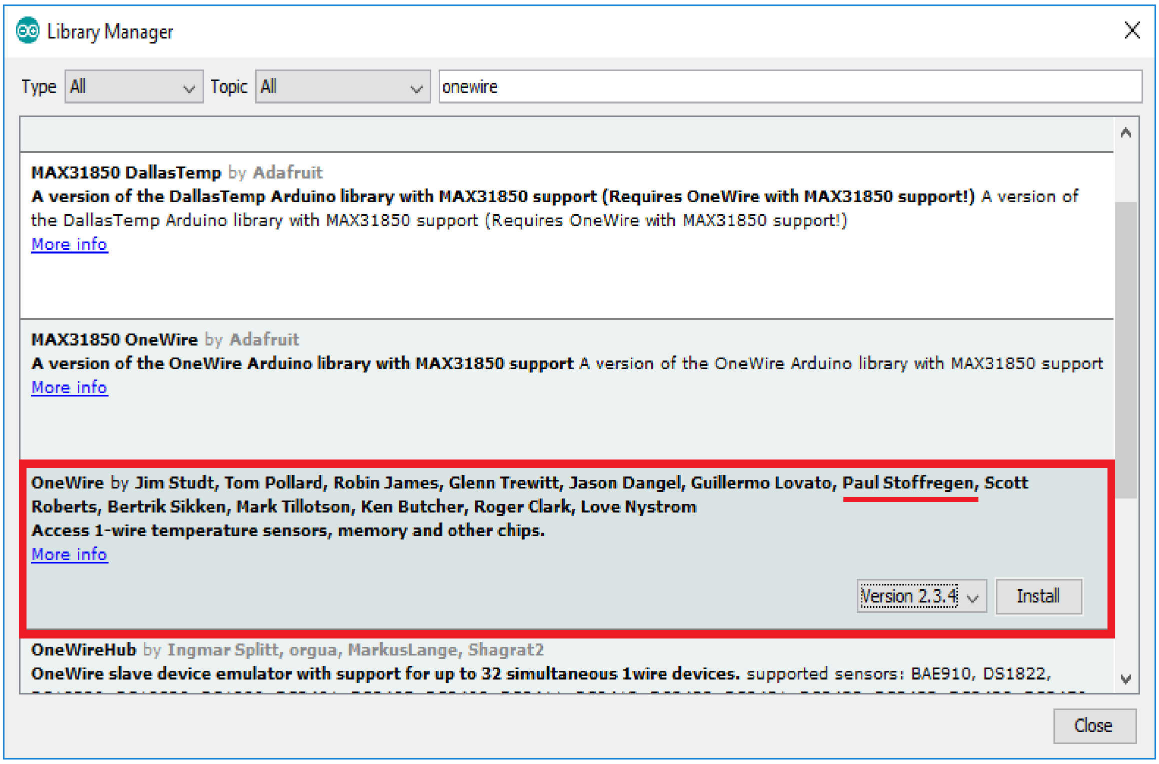
Upload Code – Single DS18B20

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.



Library Manager

Type All Topic All onewire

MAX31850 DallasTemp by Adafruit
A version of the DallasTemp Arduino library with MAX31850 support (Requires OneWire with MAX31850 support!) A version of the DallasTemp Arduino library with MAX31850 support (Requires OneWire with MAX31850 support!)
More info

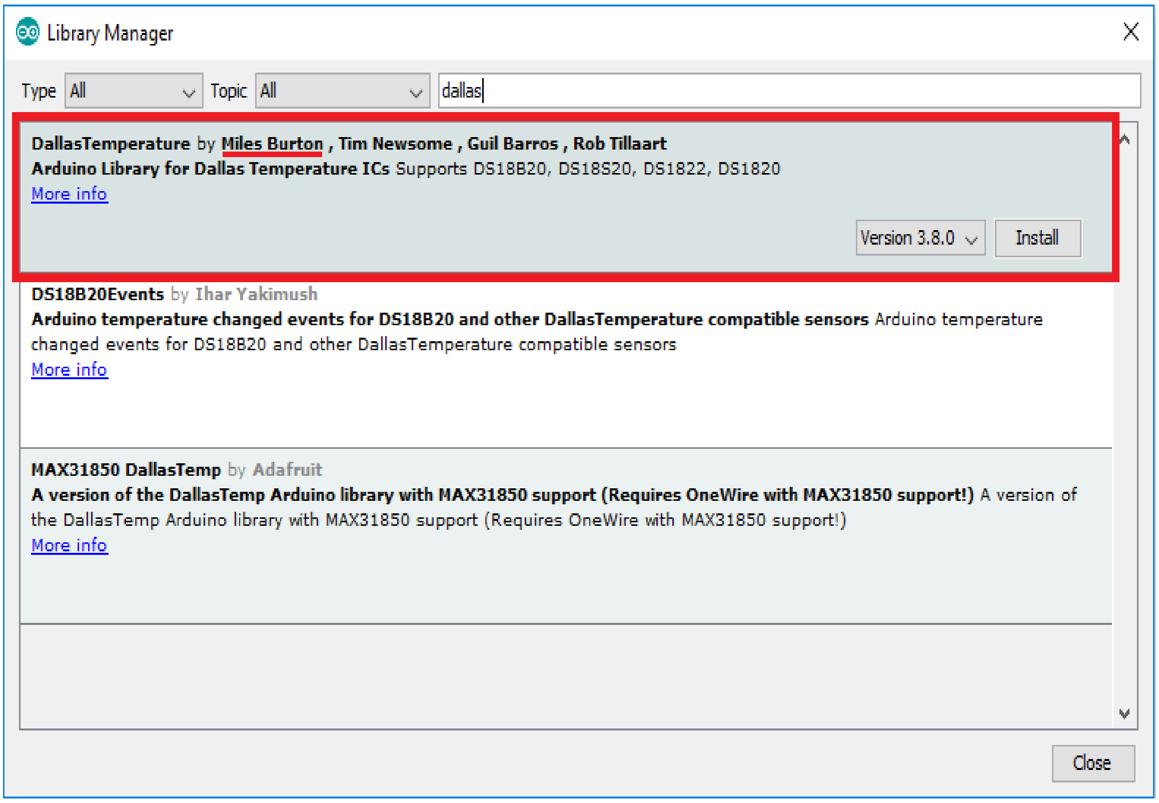
MAX31850 OneWire by Adafruit
A version of the OneWire Arduino library with MAX31850 support A version of the OneWire Arduino library with MAX31850 support
More info

OneWire by **Jim Studt, Tom Pollard, Robin James, Glenn Trewitt, Jason Dangel, Guillermo Lovato, Paul Stoffregen, Scott Roberts, Bertrik Sikken, Mark Tillotson, Ken Butcher, Roger Clark, Love Nystrom**
Access 1-wire temperature sensors, memory and other chips.
More info
Version 2.3.4 Install

OneWireHub by Ingmar Splitt, orgua, MarkusLange, Shagrat2
OneWire slave device emulator with support for up to 32 simultaneous 1wire devices. supported sensors: BAE910, DS1822,

Close

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



DallasTemperature by Miles Burton , Tim Newsome , Guil Barros , Rob Tillaart
Arduino Library for Dallas Temperature ICs Supports DS18B20, DS18S20, DS1822, DS1820
More info

DS18B20Events by Ihar Yakimush
Arduino temperature changed events for DS18B20 and other DallasTemperature compatible sensors Arduino temperature changed events for DS18B20 and other DallasTemperature compatible sensors
More info

MAX31850 DallasTemp by Adafruit
A version of the DallasTemp Arduino library with MAX31850 support (Requires OneWire with MAX31850 support!) A version of the DallasTemp Arduino library with MAX31850 support (Requires OneWire with MAX31850 support!)
More info

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.

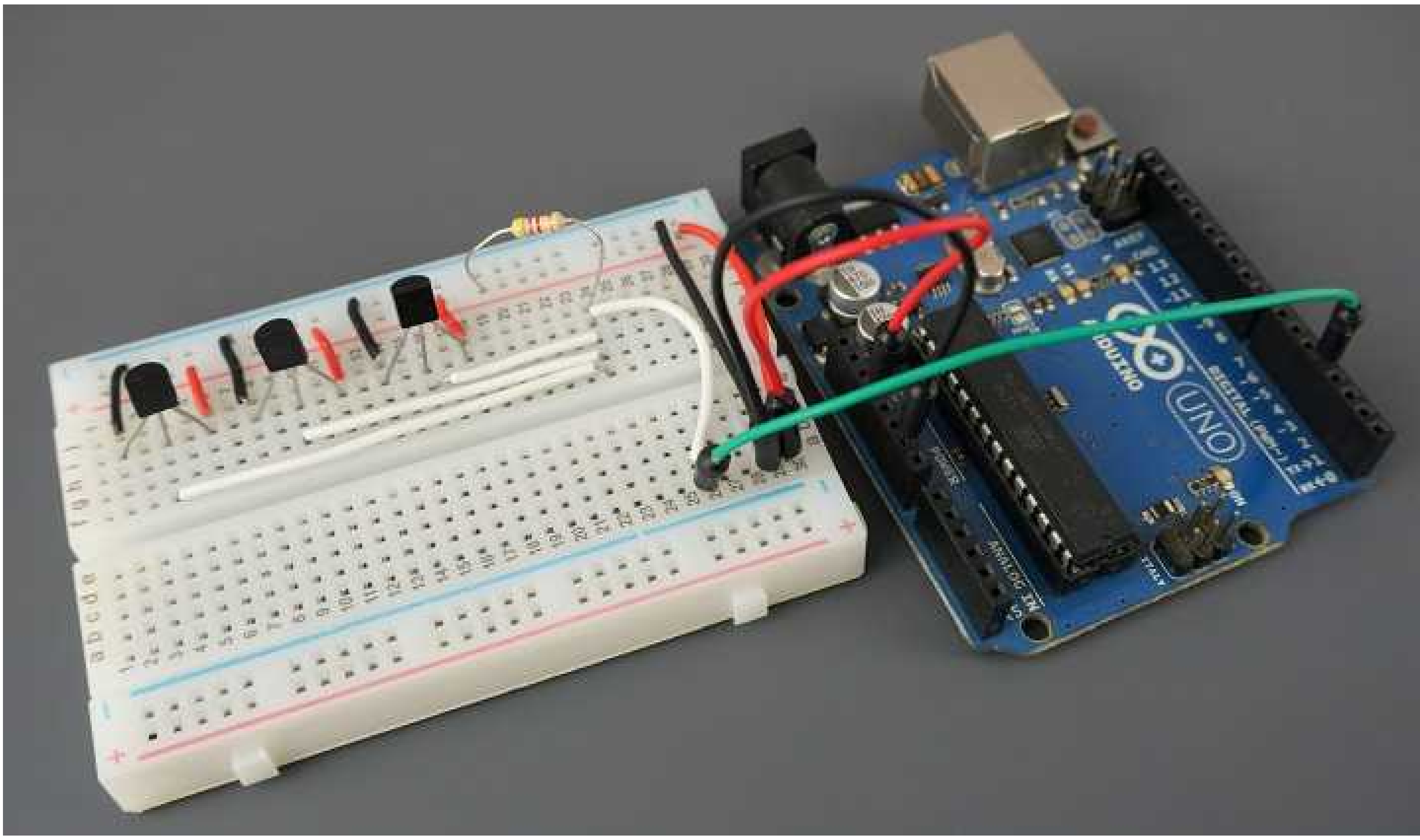
```
/*********
  Rui Santos
  Complete project details at https://randomnerdtutorials.com
  Based on the Dallas Temperature Library example
*********/

#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
  Serial.begin(9600);
  // Start up the library
  sensors.begin();
}

void loop(void){
  // Call sensors.requestTemperatures() to issue a global temperature and Requests to all devices on the bus
  sensors.requestTemperatures();

  Serial.print("Celsius temperature: ");
  // Why "byIndex"? You can have more than one IC on the same bus. 0 refers to the first IC on the wire
  Serial.print(sensors.getTempCByIndex(0));
  Serial.print(" - Fahrenheit temperature: ");
  Serial.println(sensors.getTempFByIndex(0));
  delay(5000);
}
```

Raw code

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.

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.

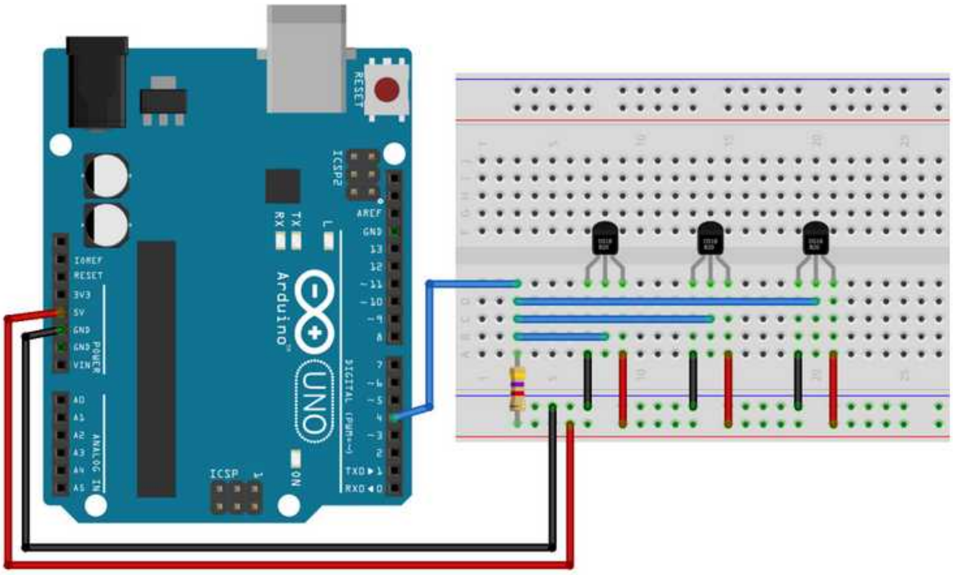
Getting Temperature from Multiple DS18B20 Sensors



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 http://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 (not just
Maxim/Dallas temperature ICs)
OneWire oneWire(ONE_WIRE_BUS);

// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);

int numberOfDevices; // Number of temperature devices found

DeviceAddress tempDeviceAddress; // We'll use this variable to store a found
device address

void setup(void) {
  // start serial port
  Serial.begin(9600);

  // Start up the library
  sensors.begin();

  // Grab a count of devices on the wire
  numberOfDevices = sensors.getDeviceCount();

  // locate devices on the bus
  Serial.print("Locating devices...");
  Serial.print("Found ");
  Serial.print(numberOfDevices, DEC);
  Serial.println(" devices.");

  // Loop through each device, print out address
  for(int i=0;i<numberOfDevices; i++) {
    // Search the wire for address
    if(sensors.getAddress(tempDeviceAddress, i)) {
      Serial.print("Found device ");
      Serial.print(i, DEC);
      Serial.print(" with address: ");
      printAddress(tempDeviceAddress);
      Serial.println();
                } else {
                  Serial.print("Found ghost device at ");
                  Serial.print(i, DEC);
                  Serial.print(" but could not detect address. Check power and
cabling");
                }
  }
}

void loop(void) {
  sensors.requestTemperatures(); // Send the command to get temperatures

  // Loop through each device, print out temperature data
  for(int i=0;i<numberOfDevices; i++) {
    // Search the wire for address
    if(sensors.getAddress(tempDeviceAddress, i)){

                // Output the device ID
                Serial.print("Temperature for device: ");
                Serial.println(i,DEC);

    // Print the data
    float tempC = sensors.getTempC(tempDeviceAddress);
    Serial.print("Temp C: ");
    Serial.print(tempC);
    Serial.print(" Temp F: ");
    Serial.println(DallasTemperature::toFahrenheit(tempC)); // Converts tempC to
Fahrenheit
    }
  }
  delay(5000);
}

// function to print a device address
void printAddress(DeviceAddress deviceAddress) {
  for (uint8_t i = 0; i < 8; i++) {
    if (deviceAddress[i] < 16) Serial.print("0");
      Serial.print(deviceAddress[i], HEX);
  }
}
```

Raw code

Source:
https://randomnerdtutorials.com/guide-for-ds18b20-temperature-sensor-with-arduino/pa