Using DS18B20 Temperature Sensor with CircuitPython

Created by Tony DiCola
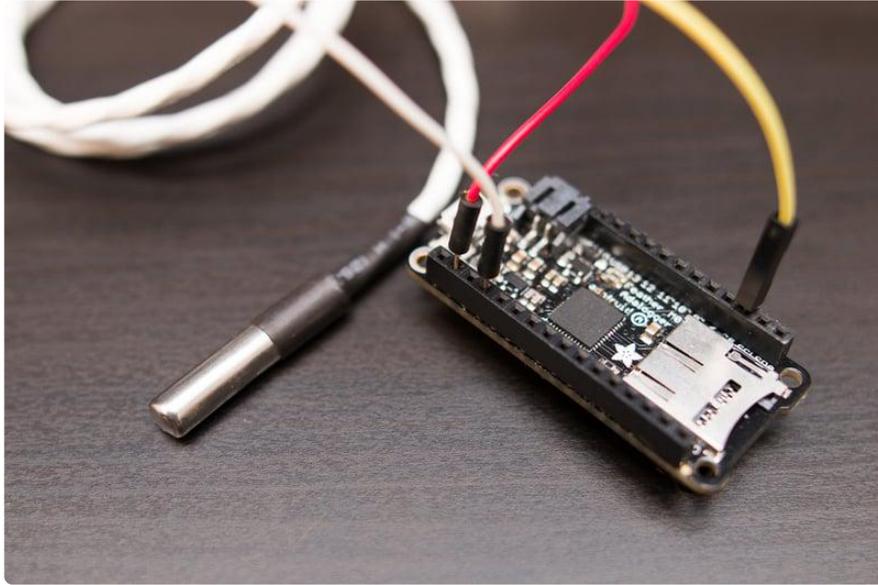

https://learn.adafruit.com/using-ds18b20-temperature-sensor-with-circuitpython
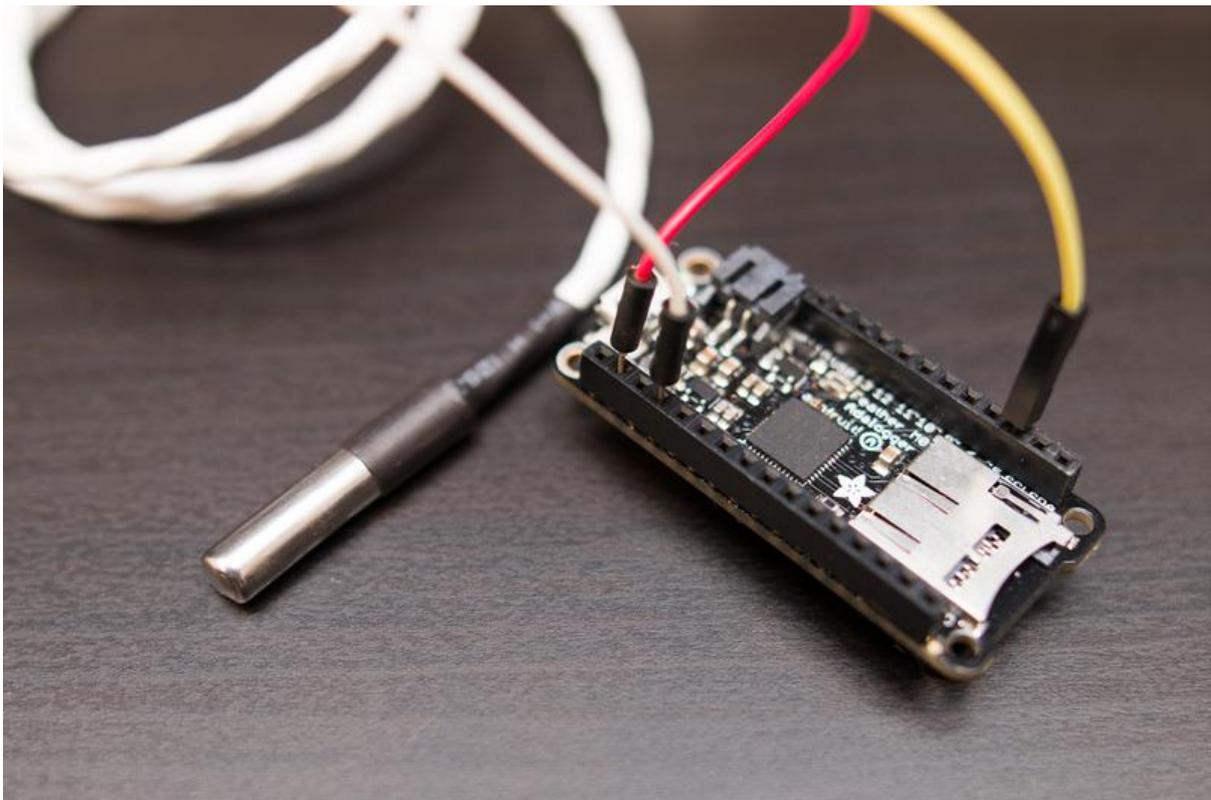
Last updated on 2022-12-01 03:09:05 PM EST

Table of Contents

Overview

One-wire temperature sensors like the DS18B20 are devices that can measure temperature with a minimal amount of hardware and wiring. These sensors use a digital protocol to send accurate temperature readings directly to your development board without the need of an analog to digital converter or other extra hardware. You can get one-wire sensors in different form factors like waterproof and high temperature probes--these are perfect for sensing temperature in many different projects and applications. And since these sensors use the one-wire protocol you can even have multiple of them connected to the same pin and read all their temperature values independently. With just a few connections and some CircuitPython code you'll be sensing temperature in no time!

This guide explores how to connect a DS18B20 one-wire temperature sensor to a CircuitPython board and read its temperature from Python code using a few simple CircuitPython modules.

Hardware

To follow this guide you'll need the following parts:

- DS18B20 Dallas one-wire temperature sensor. You can use a simple standalone [DS18B20 sensor ()](), a [waterproof DS18B20 probe sensor ()](), or a [high temperature DS18B20 probe sensor ()]().
- 4.7 KΩ Resistor. One-wire devices need a pull-up resistor connected to their signal line to be properly read by your board. Luckily all the DS18B20 sensors sold by Adafruit include the necessary pull-up resistor!
- A board running CircuitPython. You'll need a board capable of running CircuitPython like the [Feather M0 Express, Trinket M0 etc! ()]().
- [Breadboard ()]() and [jumper wires ()](). You'll need these parts to connect components to your development board.
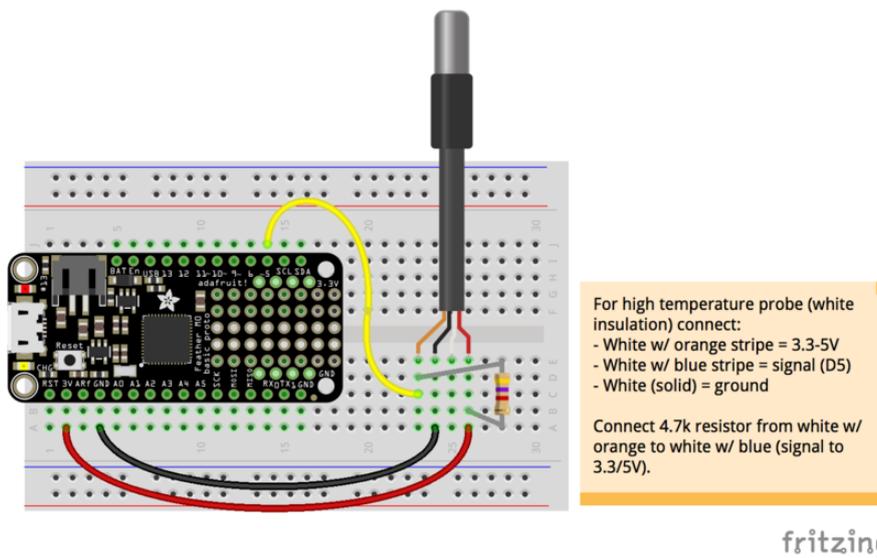
Wiring

Connect your DS18B20 sensor to a digital input on your development board. You'll also need to add the 4.7 KΩ pull-up resistor to the signal line to ensure the board can read the sensor. Here's an example of wiring a standalone DS18B20 to a Feather M0 board:
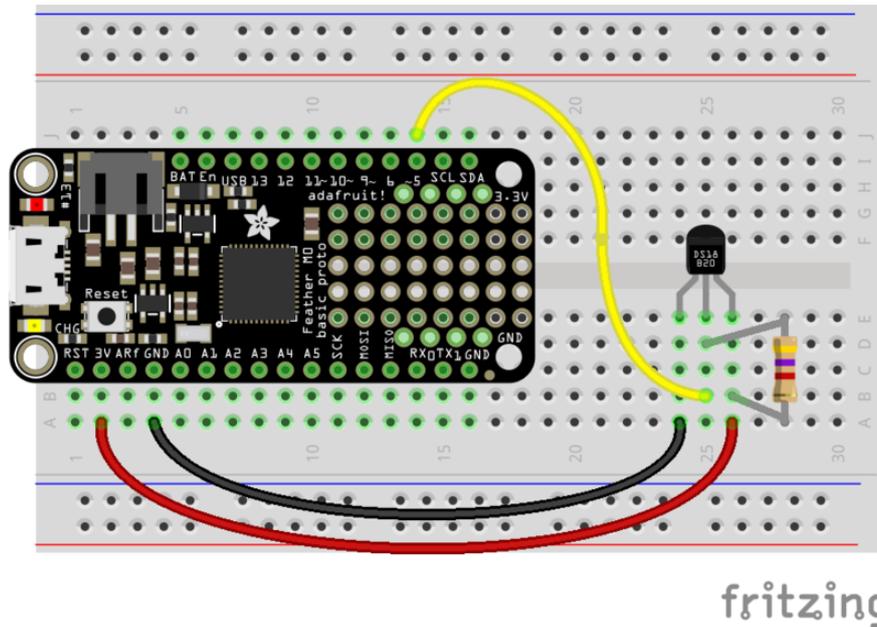
- Left-most leg of the sensor (with the flat part facing you) to board ground.
- Middle leg of the sensor to board D5.
- Right-most leg of the sensor to board 3.3V.
- 4.7 KΩ resistor connected to both the middle leg (data) and right-most leg (3.3V).

Or if you're using a waterproof or high-temperature probe here's an example of the wiring to a Feather M0 board:

For high temperature probe (white insulation) connect:
- White w/ orange stripe = 3.3-5V
- White w/ blue stripe = signal (D5)
- White (solid) = ground

Connect 4.7k resistor from white w/ orange to white w/ blue (signal to 3.3/5V).

- Black wire (or solid white on high-temperature probe) to board ground.
- Orange wire (or white with blue stripe on high-temperature probe) to board D5.
- Red wire (or white with orange stripe on high-temperature probe) to board 3.3V.

- 4.7K**Ω** resistor connected to both the data line on one side (D5 or whatever pin) and power line (3.3V) on the other side

CircuitPython

To use the DS18B20 you'll need to install both the Adafruit CircuitPython OneWire () a nd Adafruit CircuitPython DS18X20 () modules on your CircuitPython board.

First make sure you are running the latest version of Adafruit CircuitPython () for your board.

Next you'll need to install the necessary libraries to use the hardware--carefully follow the steps to find and install these libraries from Adafruit's CircuitPython library bundle (). Our introduction guide has a great page on how to install the library bundle () for both express and non-express boards.

Remember for non-express boards like the Trinket M0 or QT Py, you'll need to manually install the necessary folders and files from the bundle:

- adafruit_onewire
- adafruit_ds18x20.mpy

You can also download the adafruit_onewire folder from its releases page on Github (), and the adafruit_ds18x20.mpy from its releases page on Github ().

Before continuing make sure your board's lib folder or root filesystem has the adafruit _onewire, and adafruit_ds18x20.mpy files and folders copied over.

Next connect to the board's serial REPL ()so you are at the CircuitPython >>> prompt.

Usage

To demonstrate the usage of the sensor we'll initialize it and read the temperature from the Python REPL.

First run the following code to import the necessary modules and initialize the one-wire bus connection with the sensor:

```
import board
from adafruit_onewire.bus import OneWireBus
ow_bus = OneWireBus(board.D5)
```

Be sure to change board.D5 to the appropriate pin you're using if you've connected your sensor to a different digital input than shown on the wiring diagram of this guide!

Before you use the temperature sensor you can scan the one-wire bus to see which devices are available. This is handy for example if you have multiple sensors connected to your board and want to choose the right sensor based on its ID. Run this code to print out the ROM (or unique ID) of each sensor on the bus:

```
devices = ow_bus.scan()
for device in devices:
    print("ROM = {} \tFamily = 0x{:02x}".format([hex(i) for i in device.rom],
device.family_code))
```

```
>>> devices = ow_bus.scan()
>>> for device in devices:
...     print("ROM = {} \tFamily = 0x{:02x}".format([hex(i) for i in device.rom], device.family_code))
...
ROM = ['0x28', '0xff', '0x60', '0xe', '0xc1', '0x16', '0x5', '0xcd']     Family = 0x28
ROM = ['0x28', '0x71', '0x83', '0x5e', '0x6', '0x0', '0x0', '0x28']     Family = 0x28
```

If you only have one sensor is connected you should see one result printed with the ROM value (a list of hex values that has the unique identifier for that temperature sensor) and family (a byte that indicates the type of device).

If you multiple sensors connected you should see a line printed for each sensor. However if you don't see any devices printed double check your wiring and that the 4.7 KΩ pull-up resistor is connected as shown on the wiring diagrams!

Now you can import the DS18x20 module and use it to read the temperature from a sensor. Run the following code to import the module and create an instance of the DS18x20 sensor from the first device found on the one-wire bus:

```
import adafruit_ds18x20
ds18b20 = adafruit_ds18x20.DS18X20(ow_bus, devices[0])
```

Notice the first parameter to the DS18X20 initializer is the one-wire bus instance, and the second parameter is a device instance (found from calling scan() on the bus previously). If you have multiple sensors connected you can pick the appropriate one by changing the device instance to another value, like devices[1] would be the second sensor found (the 0 or 1 value in brackets is the index into the device list).

At this point you're ready to read the sensor's temperature property. This will return a value in degrees Celsius:

```
print('Temperature: {0:0.3f} °C'.format(ds18b20.temperature))
```

```
>>> import adafruit_ds18x20
>>> ds18b20 = adafruit_ds18x20.DS18X20(ow_bus, devices[0])
>>> print('Temperature: {0:0.3f}C'.format(ds18b20.temperature))
Temperature: 21.813C
>>>
```

Finally there's one other property you can interact with, resolution. You can read and write this property to get or set the resolution in bits of the temperature sensor. Only a value of 9, 10, 11, or 12 is supported (the default is 12-bits of resolution). For example to change to a lower 9-bit resolution:

```
ds18b20.resolution = 9
print('Resolution: {0} bits'.format(ds18b20.resolution))
print('Temperature: {0:0.3f} °C'.format(ds18b20.temperature))
```

```
>>> ds18b20.resolution = 9
>>> print('Resolution: {0} bits'.format(ds18b20.resolution))
Resolution: 9 bits
>>> print('Temperature: {0:0.3f}C'.format(ds18b20.temperature))
Temperature: 21.500C
>>>
```

That's all there is to using the DS18B20 with CircuitPython!

Here's a complete example that will print the temperature every second from the first sensor found on the bus. Save this as main.py on your board and open the serial REPL to see the output:

```
# SPDX-FileCopyrightText: 2020 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT

# Simple demo of printing the temperature from the first found DS18x20 sensor every
second.
# Author: Tony DiCola

# A 4.7Kohm pullup between DATA and POWER is REQUIRED!

import time
import board
from adafruit_onewire.bus import OneWireBus
from adafruit_ds18x20 import DS18X20


# Initialize one-wire bus on board pin D5.
ow_bus = OneWireBus(board.D5)

# Scan for sensors and grab the first one found.
ds18 = DS18X20(ow_bus, ow_bus.scan()[0])

# Main loop to print the temperature every second.
while True:
    print("Temperature: {0:0.3f}C".format(ds18.temperature))
    time.sleep(1.0)
```

you can hold the sensor in your fingers to heat it up, watch the values go up!

```
Adafruit CircuitPython REPL
Temperature:  23.063C
Temperature:  23.063C
Temperature:  23.125C
Temperature:  24.750C
Temperature:  26.063C
Temperature:  26.938C
Temperature:  27.125C
```

Python Docs

[Python Docs]() ()