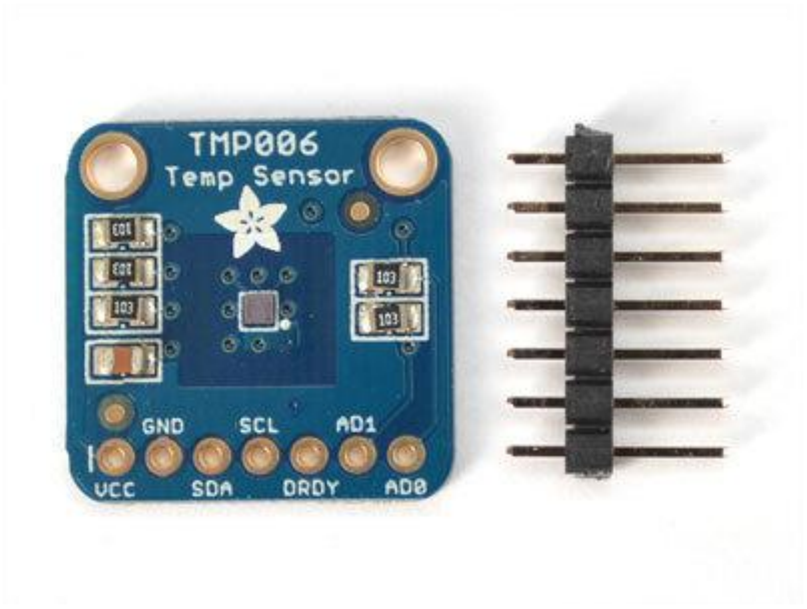
TMP006 Infrared Sensor Breakout

Created by Bill Earl

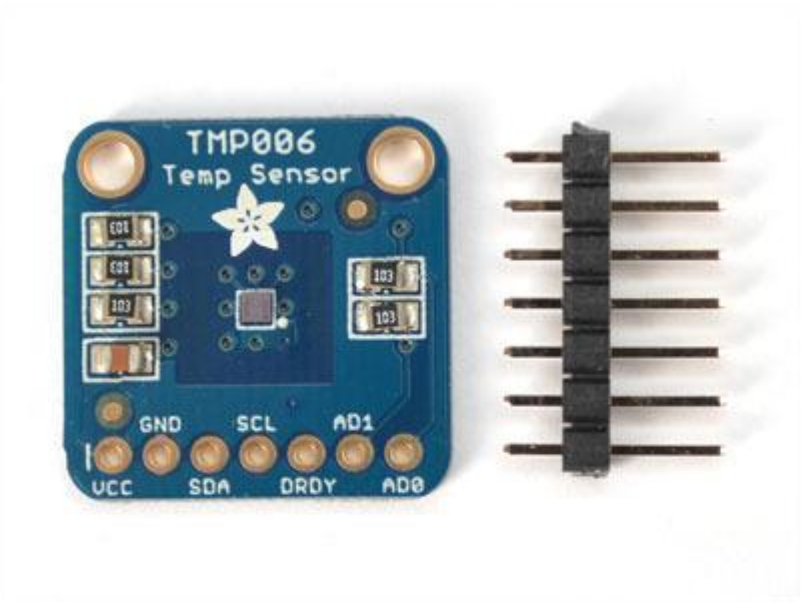


https://learn.adafruit.com/infrared-thermopile-sensor-breakout

Last updated on 2022-12-01 01:57:37 PM EST

Table of Contents

Overview



Unlike most temperature sensors, the TMP006 does not require contact with the object it is measuring. It uses a very sensitive thermopile to measure the infrared energy being emitted from the surface of the object.

This sensor works best with objects that are good emitters of infrared radiation. The ideal emitter is a completely non-reflective surface or "black body ()". Black anodized aluminum or cast iron are pretty good emitters. Polished metal surfaces are very poor emitters, but can usually be turned into a good emitter with a bit of flat-black paint.

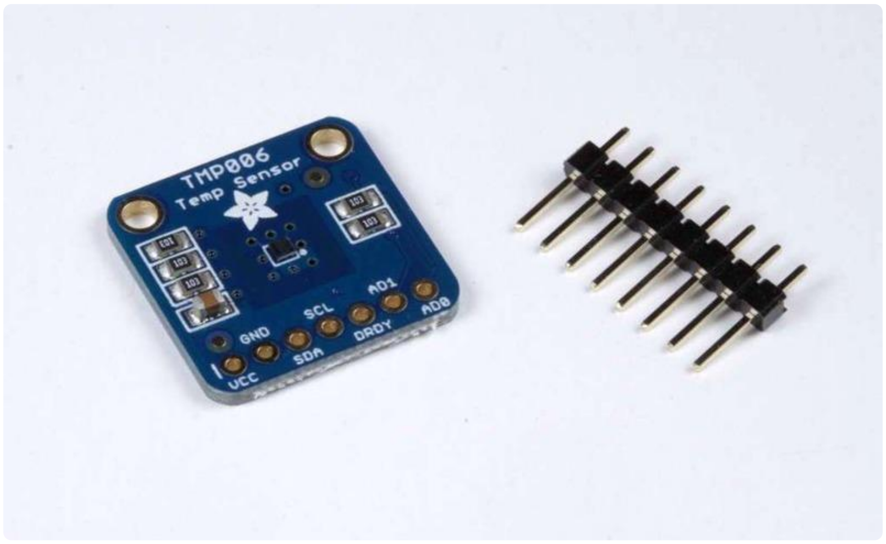
The TMP006 works with 3v to 5v, so it can be used with most microcontrollers without the need for a level shifter. It connects via the i2c bus and is addressable so you can have up to 8 TMP006 sensors on the same bus.

What is a Thermopile?

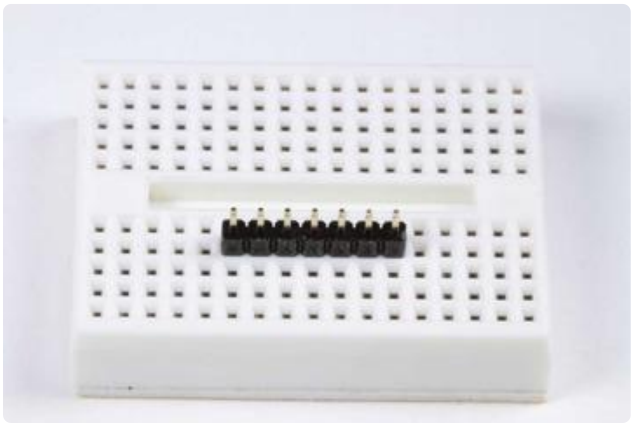
A thermopile ()is essentially a whole lot of thermocouples aligned in parallel, but wired in series. Each thermocouple will generate a microvolt-level signal proportional to the temperature differential from the hot end to the cold end. And by wiring them in parallel, the output is the sum of all outputs of all of the thermocouples. By multiplying the output like this, very small temperature differences can be measured.

Assembly and Wiring

Assembling the TMP006

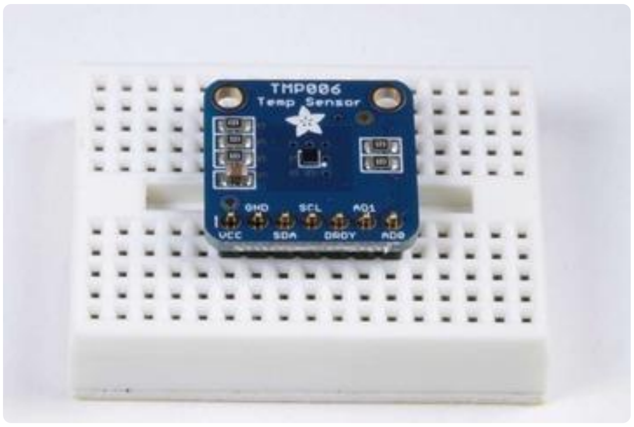


The TMP006 Thermopile Breakout comes with all surface-mount components pre-soldered. For use on a breadboard you will want to solder in the supplied header.



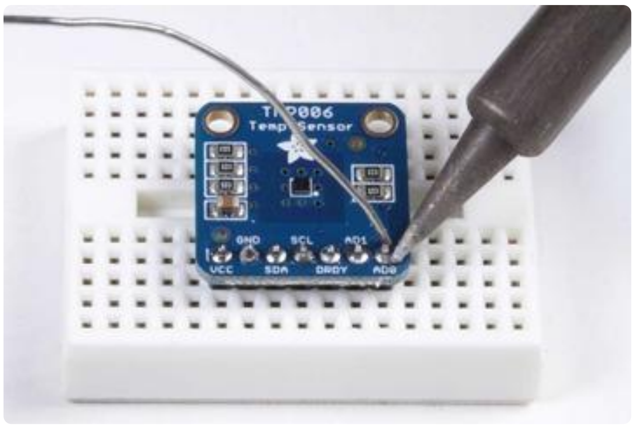
Position the Header

Cut the header to length and insert it in a breadboard with the long pins down.



Position the Breakout

Place the breakout board over the short end of the header pins.



And Solder!

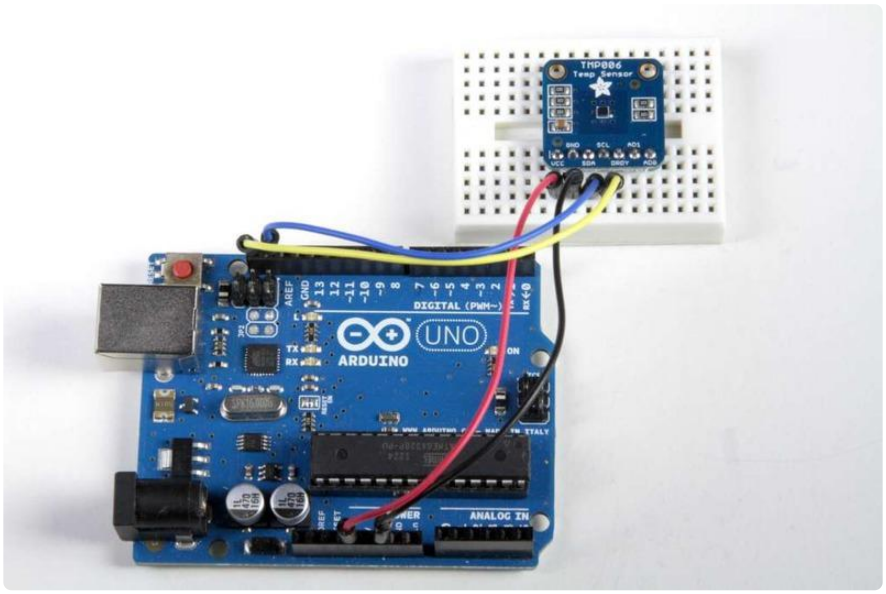
Solder each pin for a solid electrical connection.

Wiring the TMP006

TheTMP006 communicates over I2C, so you only need 4 wires to connect it to the Arduino. The SDA and SCL pins can be shared with other I2C devices.

Connections for R3 and Later Arduinos:

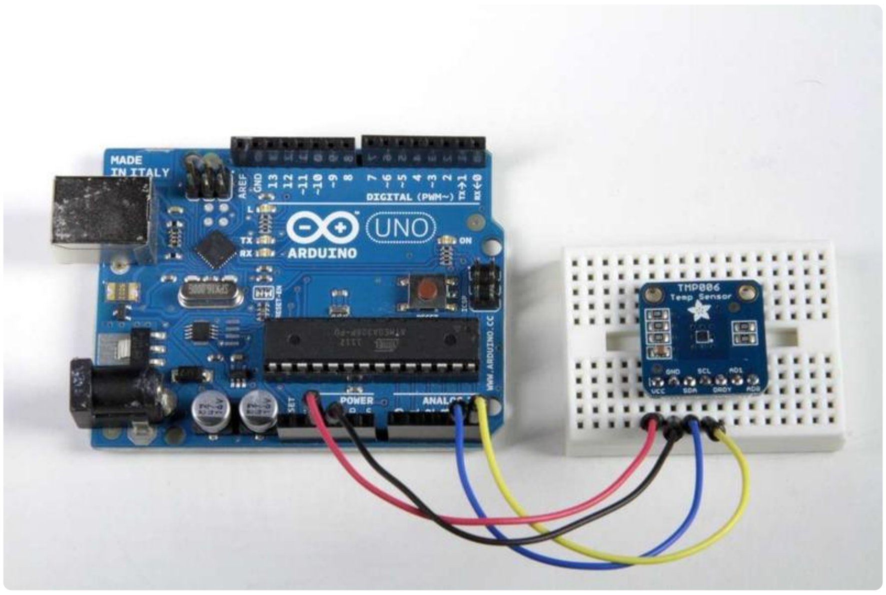
- VCC -> 3.3v or 5v
- GND -> GND
- SDA -> SDA
- SCL -> SCL



Connections for 'Classic" Arduinos:

- VCC -> 3.3v or 5v
- GND -> GND
- SDA -> Analog 4 (20 on Mega)

- SCL -> Analog 5 (21 on Mega)



Addressing

To use multiple TMP006 breakouts on one i2c bus, you need to give each one a different i2c address. The chip uses a clever addressing scheme to allow up to 8 addresses using just 2 address pins.

The default address of the TMP006 is 0x40. By connecting the address pins as in the following table, you can generate any address between 0x40 and 0x47

ADO AD1 i2c Address GND GND 0x40 GND VCC 0x41 GND SDA 0x42 GND SCL 0x43 VCC GND 0x44 VCC VCC 0x45 VCC SDA 0x46 VCC SCL 0x47

For extra credit:

The address lines are sampled continuously. By clever manipulation of the address lines, it is possible to dynamically re-address TMP006 breakouts to expand the total number of breakouts connected to the bus.

Using the Thermopile Sensor

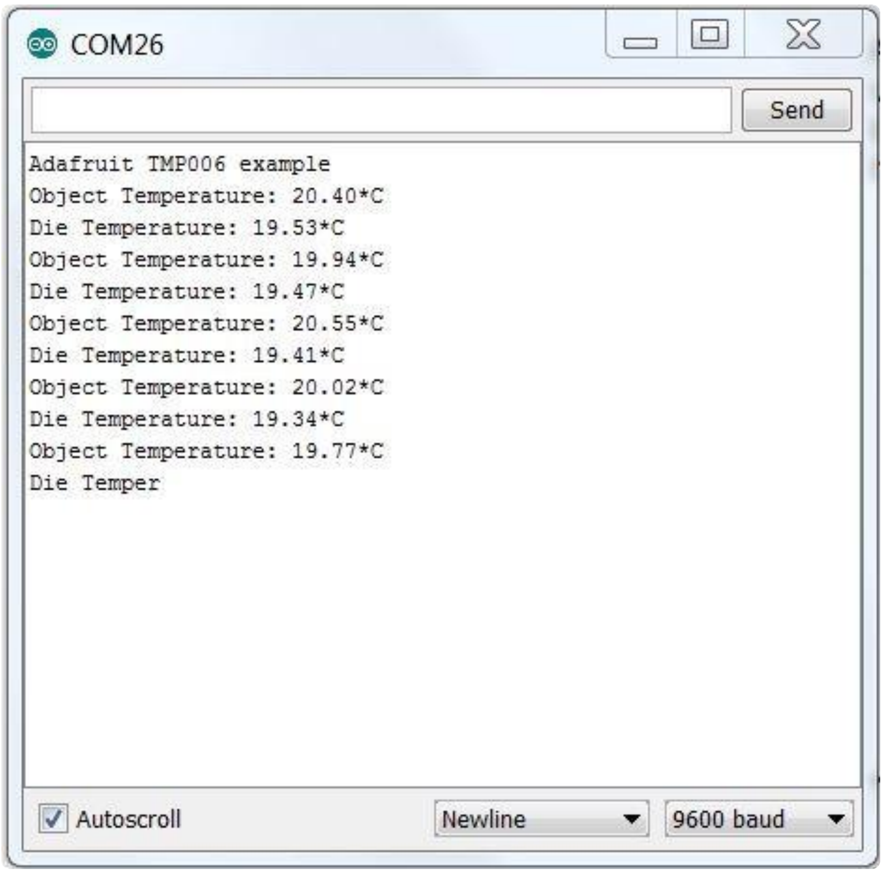
Download the Library

First download and install the TMP006 sensor library (). For tips on installing libraries, see our All About Arduino Libraries Guide ().

Compile the Example Code

Open the tmp006 example sketch in the Arduino IDE, compile and upload it. This will verify that you have the library installed correctly and let you start experimenting with basic sensor readings.

Once you upload the example sketch, open the Serial Monitor and view the output. The sensor should display a new reading about every 4 seconds.



```
Adafruit TMP006 example
Object Temperature: 20.40*C
Die Temperature: 19.53*C
Object Temperature: 19.94*C
Die Temperature: 19.47*C
Object Temperature: 20.55*C
Die Temperature: 19.41*C
Object Temperature: 20.02*C
Die Temperature: 19.34*C
Object Temperature: 19.77*C
Die Temper
```

Sensor Applications

There are many advantages to contact-less temperature sensing. But there are a few important things to consider when using these sensors. We will cover a couple of the most important ones here. For more detail, you will want to read through the TMP006 User's Guide ().

Angle of View

An important thing to consider for a contact-less sensor aplications is the angle of view. The TMP006 has a very wide angle of view. If you are not careful in positioning the sensor relative to the surface being measured, the readings will be influenced by other nearby surfaces.

The following video from TI demonstrates the angle of view and how to control it.

Surface to be Measured

As mentioned earlier, the sensor works best when measuring a surface that is a good emitter. In general, that means a dull, dark colored surface. You will not get good readings from shiny, reflective objects.

There is a handy table of emmisivities of common objects here: http://www.engineeringtoolbox.com/emissivity-coefficients-d_447.html ()

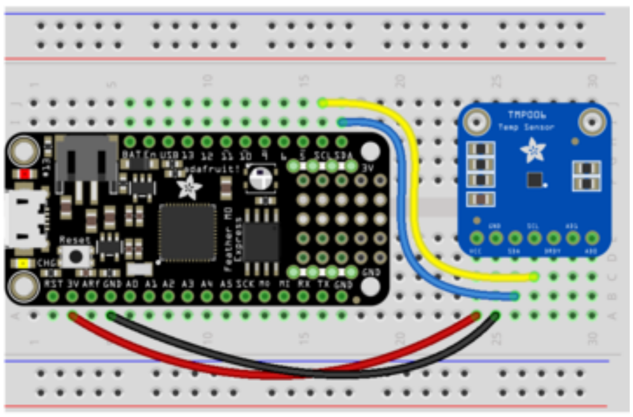
Python & CircuitPython

It's easy to use the TMP006 sensor with Python and CircuitPython, and the Adafruit CircuitPython TMP006 () module. This module allows you to easily write Python code that reads the temperature from the sensor.

You can use this sensor with any CircuitPython microcontroller board or with a computer that has GPIO and Python thanks to Adafruit_Blinka, our CircuitPython-for-Python compatibility library ().

CircuitPython Microcontroller Wiring

First, wire up a TMP006 to your board exactly as shown on the previous pages for Arduino. Here's an example of wiring a Feather M0 to the sensor with I2C:

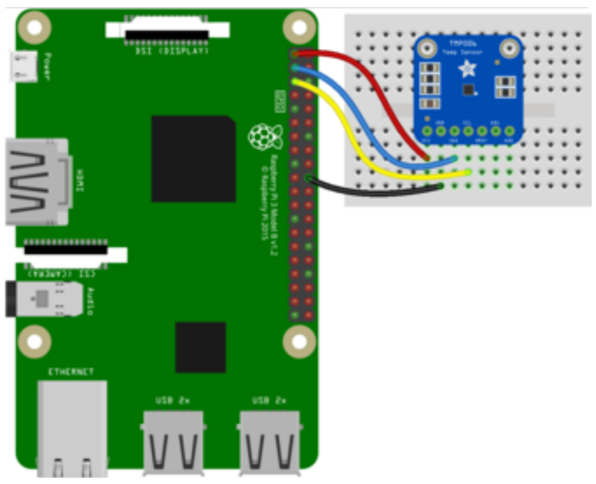


Board 3V to sensor VCC
Board GND to sensor GND
Board SCL to sensor SCL
Board SDA to sensor SDA

Python Computer Wiring

Since there's dozens of Linux computers/boards you can use, we will show wiring for Raspberry Pi. For other platforms, please visit the guide for CircuitPython on Linux to see whether your platform is supported ().

Here's the Raspberry Pi wired with I2C:



Pi 3V3 to sensor VCC
Pi GND to sensor GND
Pi SCL to sensor SCL
Pi SDA to sensor SDA

CircuitPython Installation of TMP006 Library

Next, you'll need to install the Adafruit CircuitPython TMP006 () library 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.

You need to copy the following libraries out of the library bundle into the lib folder on your CIRCUITPY drive:

- adafruit_tmp006.mpy
- adafruit_bus_device

Before continuing make sure your board's lib folder or root filesystem has the adafruit_tmp006.mpy, and adafruit_bus_device files and folders copied over.

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

Python Installation of TMP006 Library

You'll need to install the Adafruit_Blinka library that provides the CircuitPython support in Python. This may also require enabling I2C on your platform and verifying you are running Python 3. Since each platform is a little different, and Linux changes often, please visit the CircuitPython on Linux guide to get your computer ready ()!

Once that's done, from your command line run the following command:

- `sudo pip3 install adafruit-circuitpython-tmp006`

If your default Python is version 3 you may need to run 'pip' instead. Just make sure you aren't trying to use CircuitPython on Python 2.x, it isn't supported!

CircuitPython & Python Usage

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

Run the following code to import the necessary modules and initialize the I2C connection with the sensor:

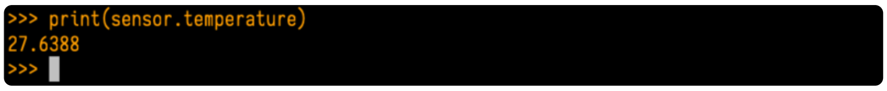
```
import board
import busio
import adafruit_tmp006

i2c = busio.I2C(board.SCL, board.SDA)
sensor = adafruit_tmp006.TMP006(i2c)
```

Now you're ready to read values from the sensor using the following property:

- temperature - The object temperature in degrees Celsius.

```
print(sensor.temperature)
```



```
>>> print(sensor.temperature)
27.6388
>>> 
```

That's all there is to reading object temperature with the TMP006 and CircuitPython!

Full Example Code

```
# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT

import time
import board
import busio
import adafruit_tmp006

# Define a function to convert celsius to fahrenheit.
def c_to_f(c):
    return c * 9.0 / 5.0 + 32.0


# Create library object using our Bus I2C port
i2c = busio.I2C(board.SCL, board.SDA)
sensor = adafruit_tmp006.TMP006(i2c)

# Initialize communication with the sensor, using the default 16 samples per
conversion.
# This is the best accuracy but a little slower at reacting to changes.
# The first sample will be meaningless
while True:
    obj_temp = sensor.temperature
    print(
        "Object temperature: {0:0.3F}*C / {1:0.3F}*F".format(obj_temp,
c_to_f(obj_temp))
    )
    time.sleep(5.0)
```

Python Docs

Python Docs ()

Downloads

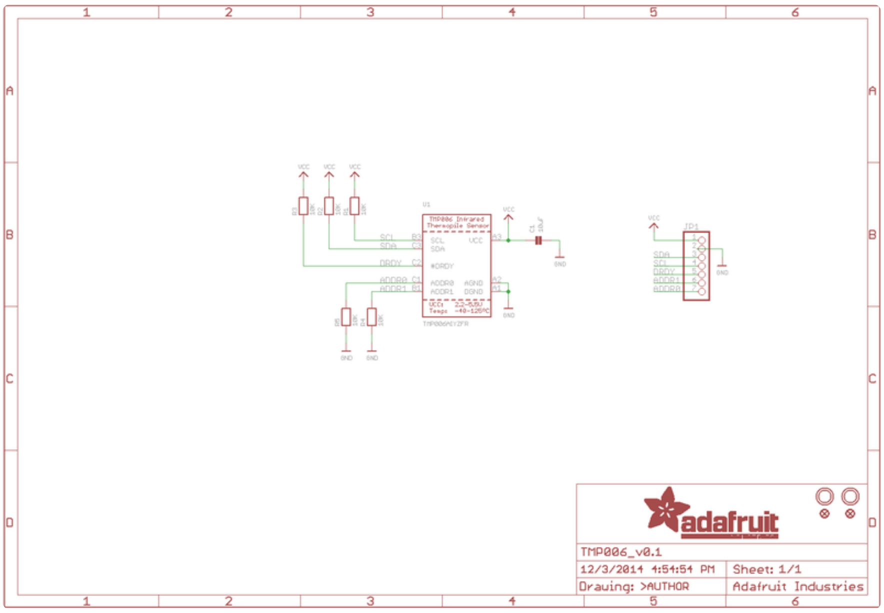
Adafruit's TMP006 Library for the Arduino. ()

Files

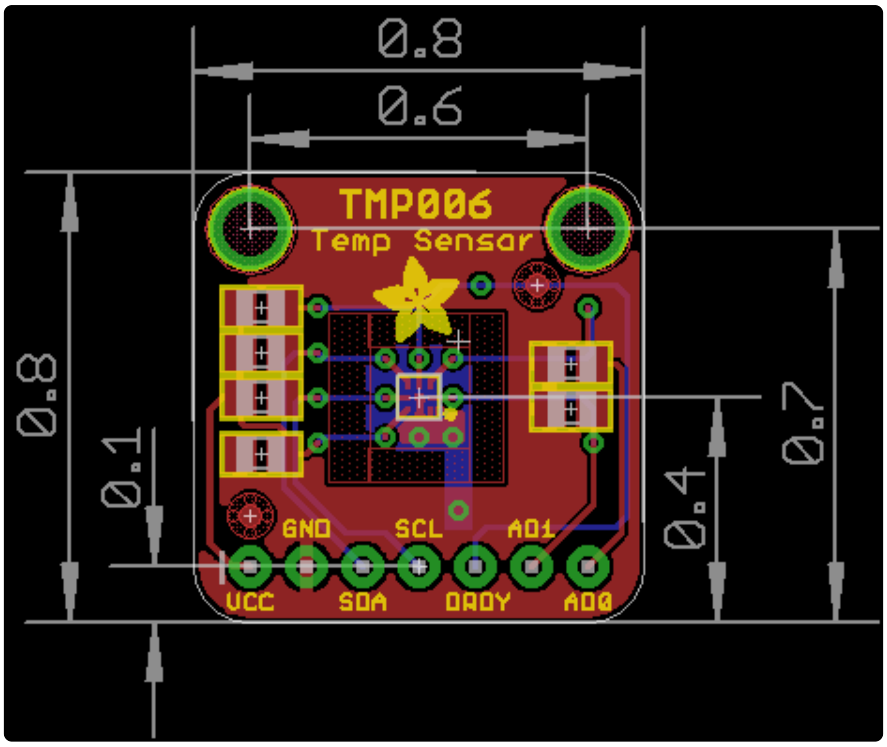
- TMP006 Data Sheet () from TI
- TMP006 User's Guide () from TI
- Fritzing object in Adafruit Fritzing Library ()

- EagleCAD PCB files ()

Schematic



Fabrication Print



0.8
0.6
TMP006
Temp Sensor
GND
SCL
AD1
VCC
SDA
DRDY
AD0
0.8
0.1
0.4
0.7