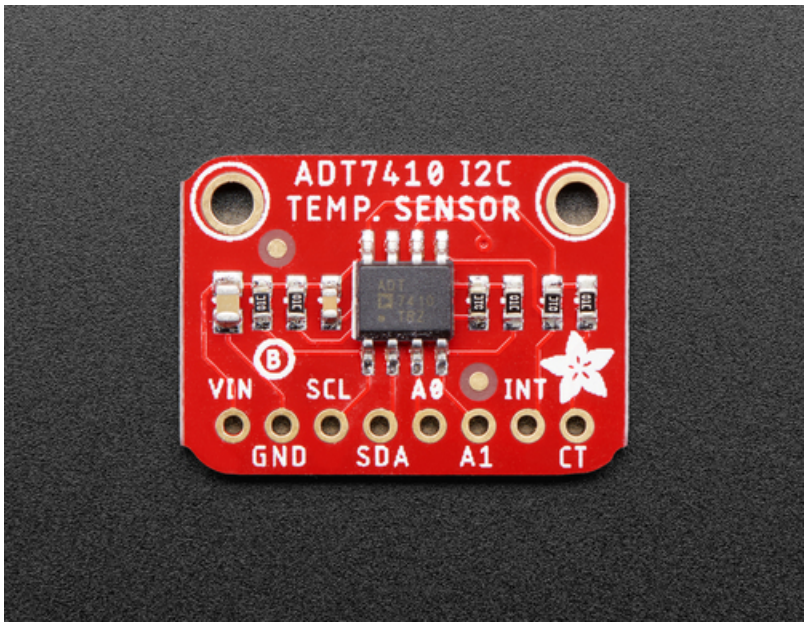
adafruit learning system

Analog Devices ADT7410 Breakout

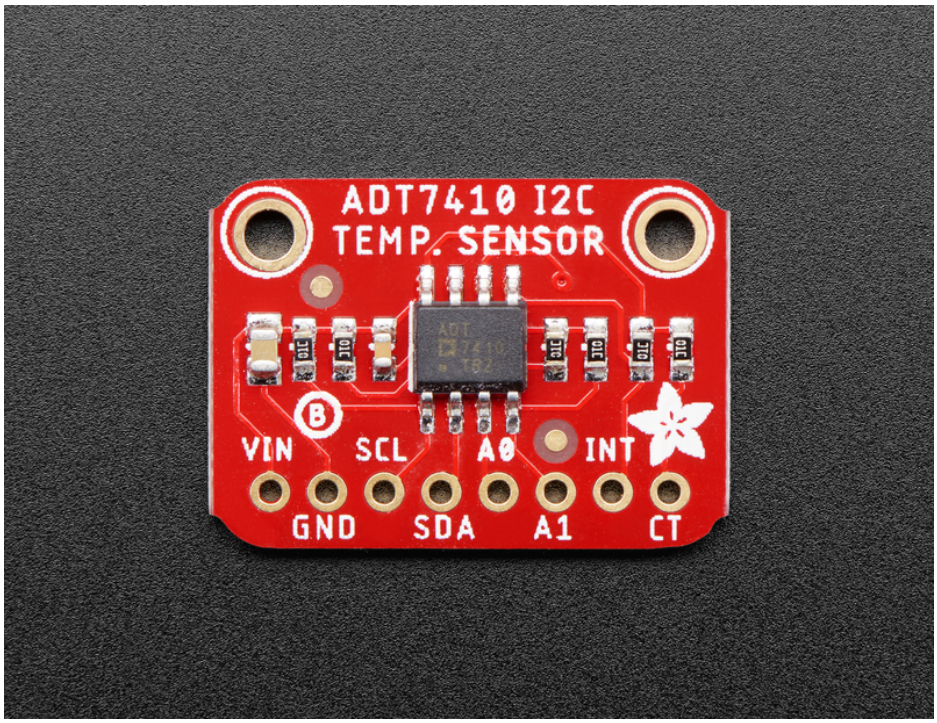
Created by Brent Rubell



Last updated on 2019-02-15 01:47:36 AM UTC

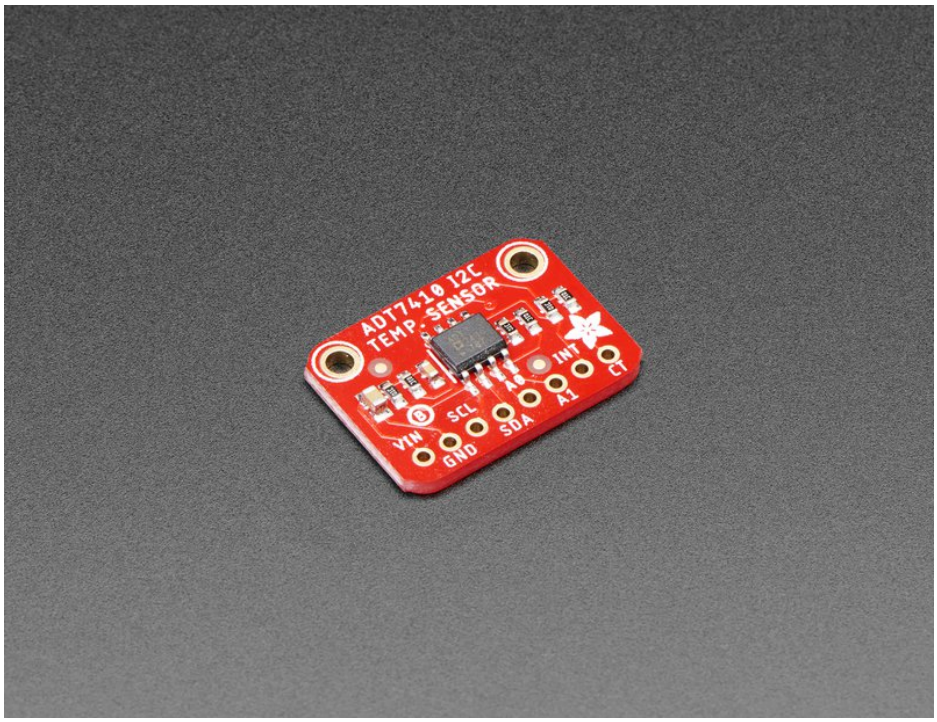
Guide Contents

Overview



Analog Devices, known for their reliable and well-documented sensor chips - has a high precision and high resolution temperature sensor on the market, and we've got a breakout to make it easy to use! The **Analog Devices ADT7410** gets straight to the point - it's an I2C temperature sensor, with 16-bit 0.0078°C temperature resolution and 0.5°C temperature tolerance. Wire it up to your microcontroller or single-board computer to get reliable temperature readings with ease

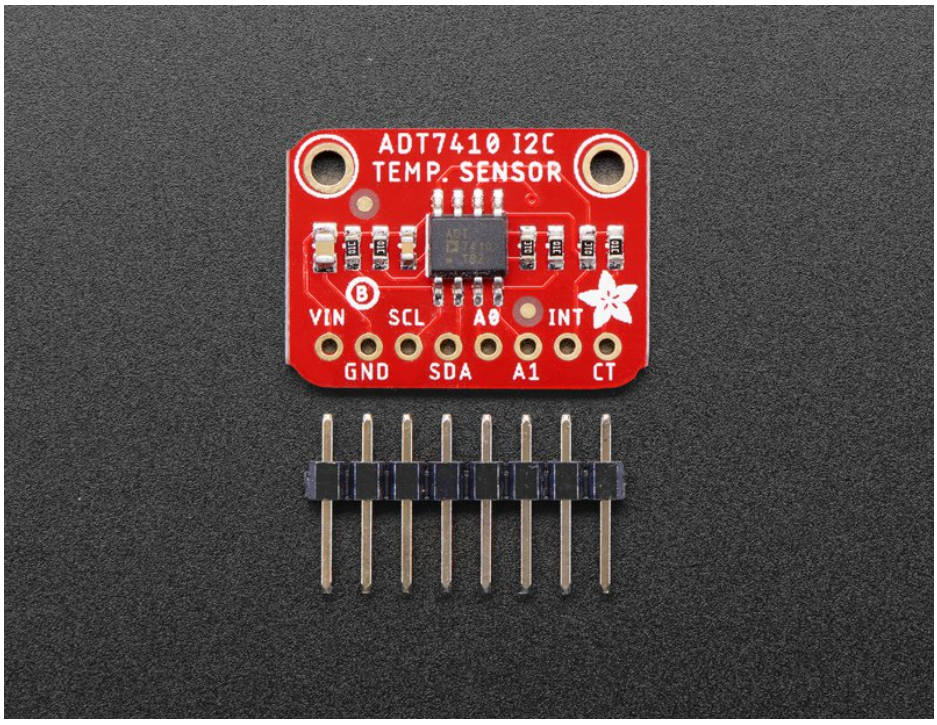
The ADT7410 has 2 address pins, so you can have up to 4 sensors on one I2C bus. There's also interrupt and critical-temperature alert pins. The sensor is good from 2.7V to 5.5V power and logic, for easy integration.



We've got both Arduino (C/C++) and CircuitPython (Python 3) libraries available so you can use it with any microcontroller like Arduino, ESP8266, Metro, etc or with Raspberry Pi or other Linux computers thanks to Blinka (our CircuitPython library support helper) (https://adafru.it/BSN).

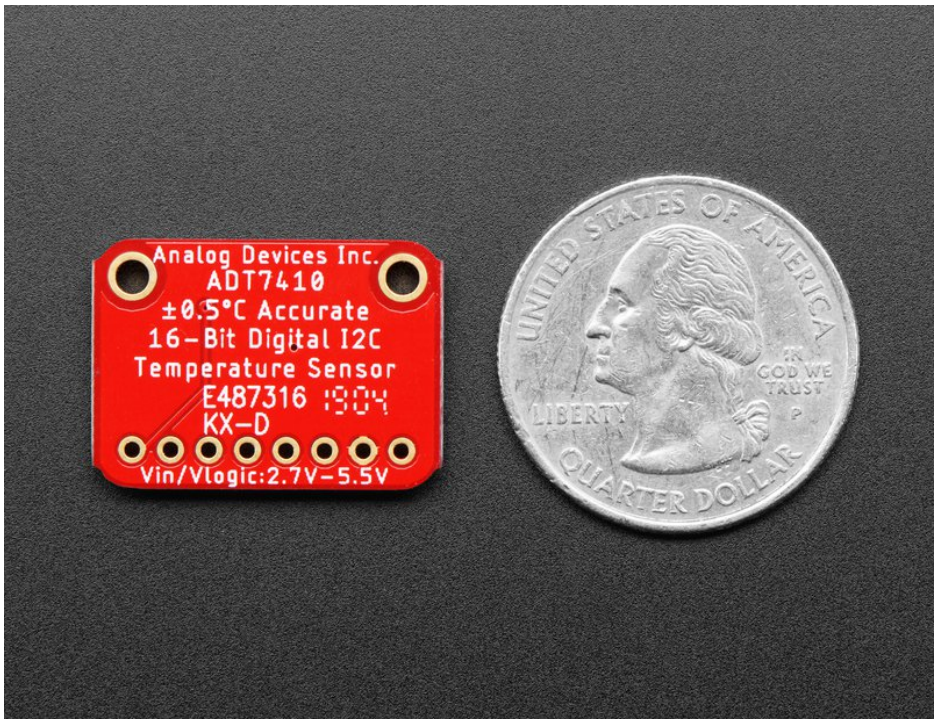
Each order comes with a fully tested and assembled breakout and some header for soldering to a PCB or breadboard. You'll be up and running in under 5 minutes!

Thanks to Digi-Key (https://adafru.it/BJr) and Analog Devices (https://adafru.it/DPF) for sponsoring the development of this breakout board - we've made the PCB "Digi-Key red (https://adafru.it/BJr)" in their honor!

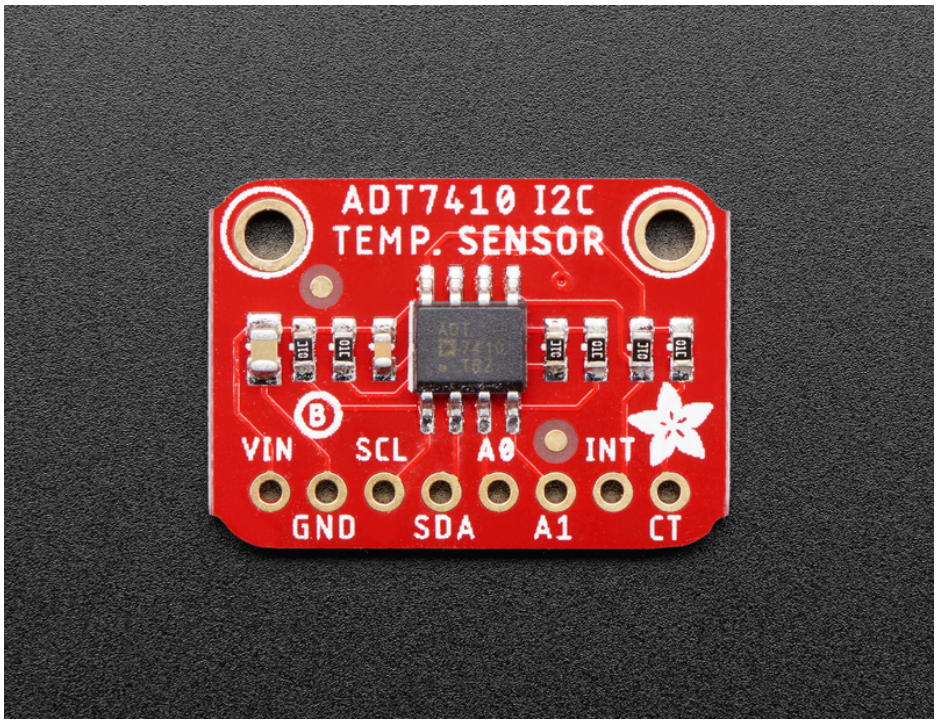


Technical Specs

- Wide input-voltage range: 2.7 V to 5.5 V
- Up to 16-bit temperature resolution (0.0078°C per lsb), default is 13 bits (0.0625°C per lsb).
- Highly accurate temperature tolerances:
    - ±0.5°C from −40°C to +105°C (2.7 V to 3.6 V)
    - ±0.4°C from −40°C to +105°C (3.0 V)
- Configurable I2C address allows up to four sensors on the I2C bus
- Operates over I2C, so only two shared lines required



Pinouts



Power Pins

- **VIN**: This is the voltage input to power for the sensor. You can connect either 5V or 3.3V to this, depending on the logic level of the MCU you are using. (Do not exceed 5V on this pin or you will permanently damage the sensor!)
- **GND**: Connect the GND pin on the breakout to a GND pin on your MCU to have a common reference point.

Digital Logic Pins

- **SCL**: This is the I2C clock line, which is pulled high to the same logic level as the **VIN** pin. Connect this to SCL on your development board.
- **SDA**: This is the I2C data line, which is pulled high to the same logic level as the **VIN** pin. Connect this to SDA on your development board.
- **A0**: This pin can be used to change the default I2C address for the sensor. See **I2C Address Options** further down for details.
- **A1**: This pin can be used to change the default I2C address for the sensor. See **I2C Address Options** further down for details.
- **INT**: This is the 'open-drain' interrupt output pin, and can be optionally connected to your MCU to trigger a HW interrupt whenever an appropriate event happens with the sensor. See the datasheet and driver for further details. It will go low or logic '0' when it is asserted.
- **CT**: This 'open-drain' pin can be configured to trigger (it will go low or to logic '0' when a**Critical Temperature (CT)** threshold is passed.

I2C Address Options

The I2C address on the ADT7410 will default to **0X48**.

If this address is used by another sensor, or if you want to enable multiple ADT7410 temperature sensors on your

device, you can change the I2C address for your specific breakout via the **A0** and **A1** pins, based on the following naming conventions:

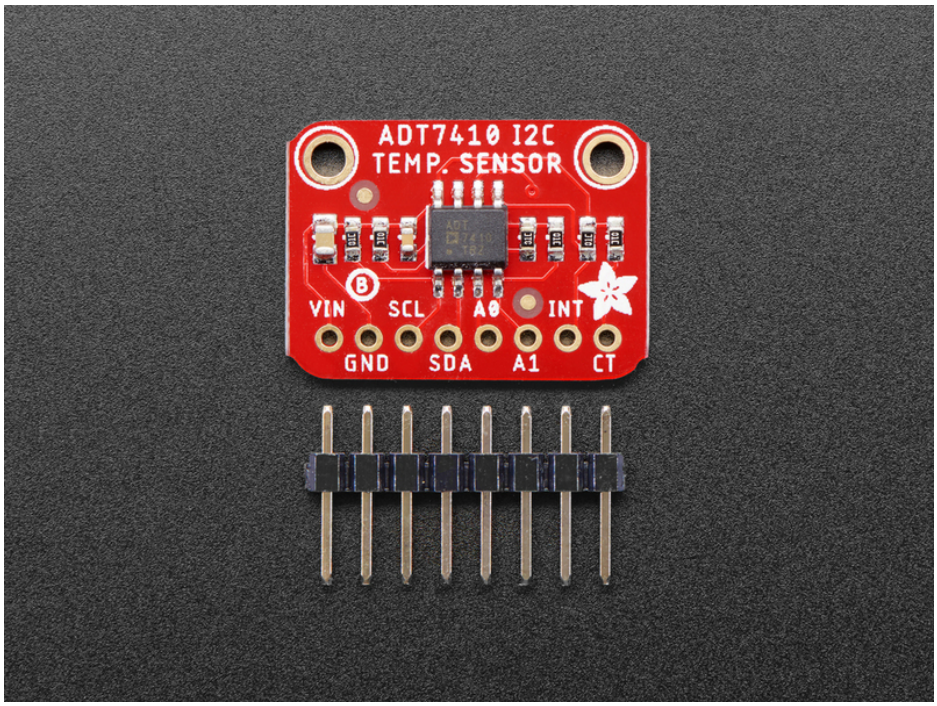
Table 20. I²C Bus Address Options

Binary							
A6	**A5**	**A4**	**A3**	**A2**	**A1**	**A0**	**Hex**
1	0	0	1	0	0	0	0x48
1	0	0	1	0	0	1	0x49
1	0	0	1	0	1	0	0x4A
1	0	0	1	0	1	1	0x4B

A0 and A1 are **pulled low** by default, which is how we end up with the default address of **0x48**.

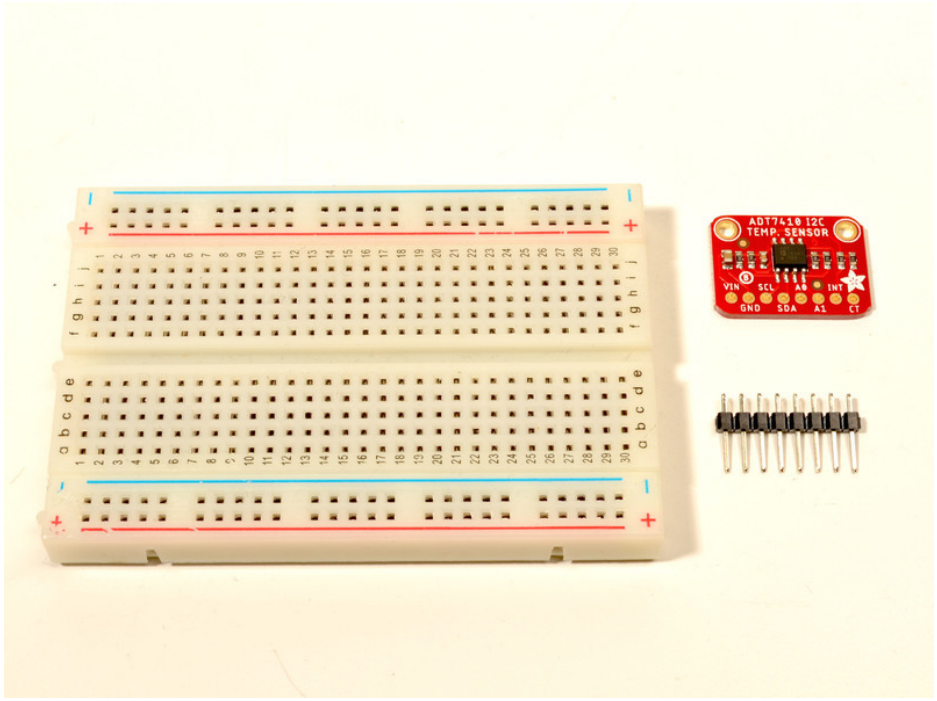
The A6..A2 bits are hard-coded and should be ignored.

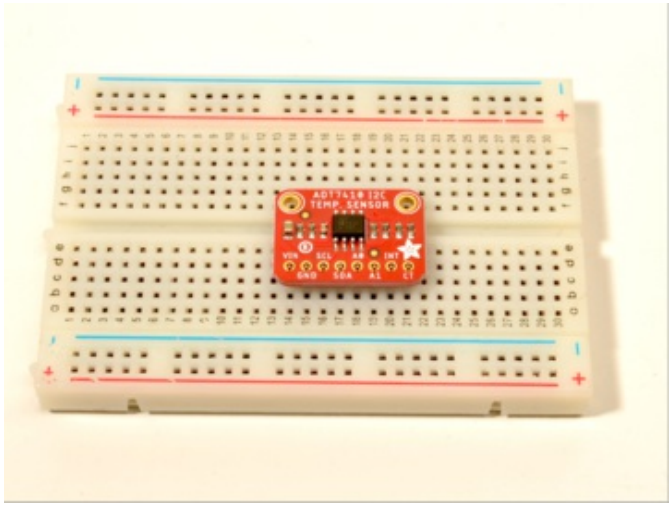
Assembly



Assembly

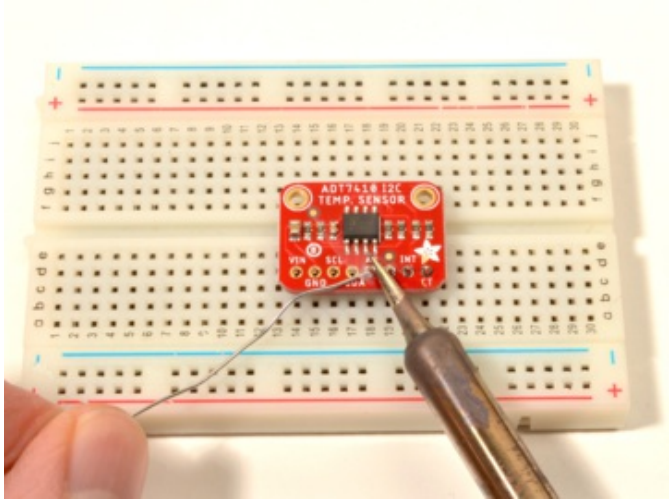
Start by preparing the header strip, cut to length if necessary. It will be easier to solder if you insert it into a breadboard - **long pins down**.





Add the ADT7410:

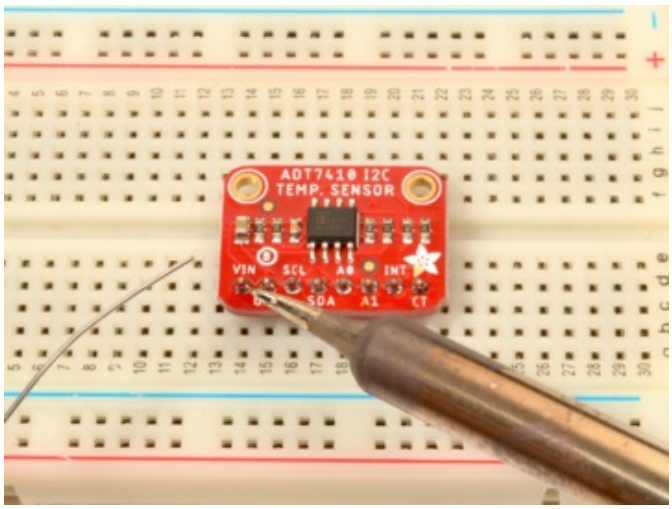
Place the board over the pins so that the short pins poke through the top of the breakout pads



And Solder!

Be sure to solder all pins for reliable electrical contact.

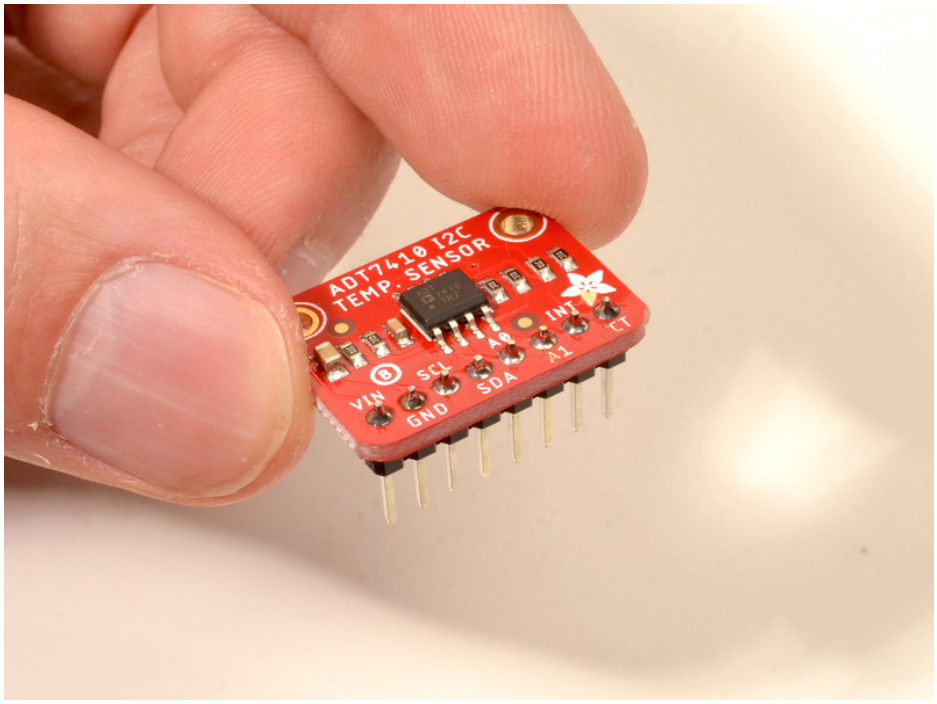
(For tips on soldering, be sure to check out the Guide to Excellent Soldering (https://adafru.it/aTk).



OK, you're done!

You can now plug in your ADT7410 breakout and enjoy using this high precision and high resolution temperature

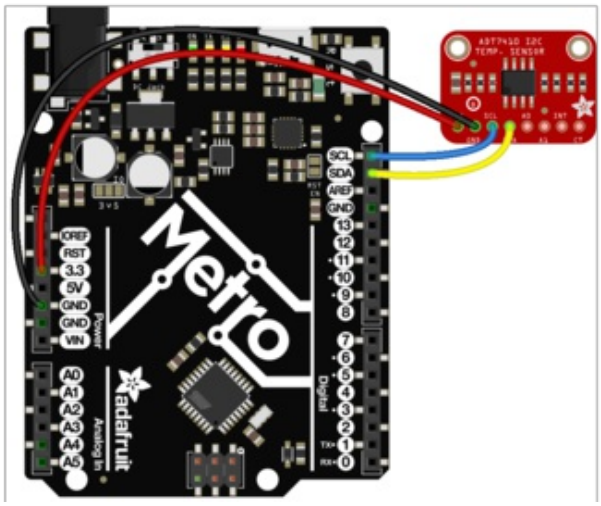
sensor.



Arduino

Arduino Wiring

You can easily wire this breakout to any Arduino-compatible microcontroller.



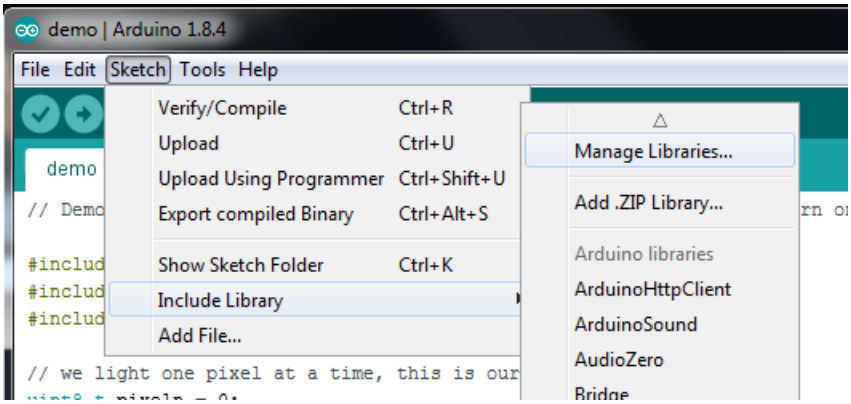
Make the following connections between the **Metro and the ADT7410**

- **ADT7410 Vin** to **3V or 5V** (depending on the logic level of your board).
- **ADT7410 GND** to **Metro GND**.
- **ADT7410 SCL** to **Metro SCL**
- **ADT7410 SDA** to **Metro SDA**

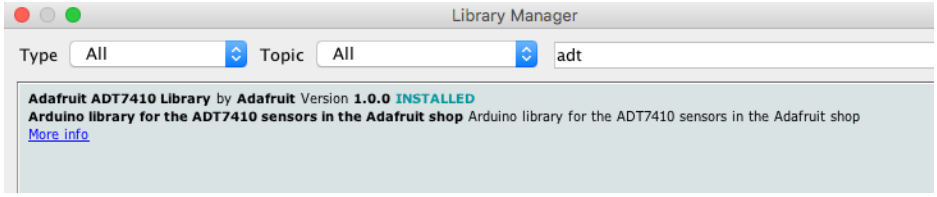
Install Adafruit_ADT7410 Library

To read data from your ADT7410, you will need to install the Adafruit_ADT7410 library (https://adafru.it/DPy). It is available from the Arduino library manager so we recommend using that.

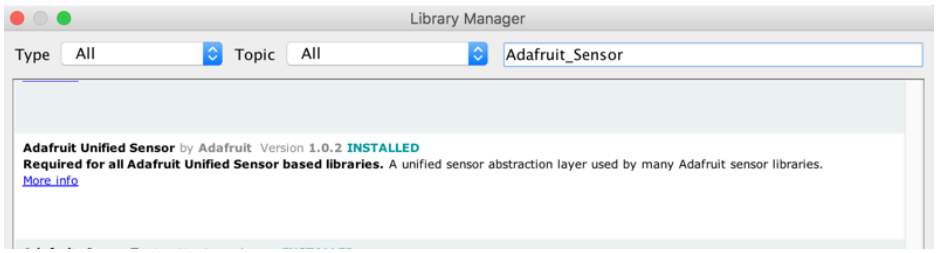
From the Arduino IDE, open the Library Manager (**Sketch -> Include Library -> Manage Libraries**)



Type in **Adafruit ADT7410** and **click Install**



You'll also need to install **Adafruit Unified Sensor Library**:



Load Example

Open up **File -> Examples -> Adafruit ADT7410 Library -> adt7410test** and upload to your Arduino wired up to the sensor

Upload the sketch to your board and open up the Serial Monitor (**Tools->Serial Monitor**). You should see the temperature in Celsius and Fahrenheit.

```
Temp: 23.25*C 73.85*F
Temp: 23.19*C 73.74*F
Temp: 23.25*C 73.85*F
Temp: 23.25*C 73.85*F
```

Library Reference

Create the ADT7410 Sensor Object:

```
Adafruit_ADT7410 tempsensor = Adafruit_ADT7410();
```

Initialize the sensor with:

```
tempsensor.begin()
```

This function returns **True** if the ADT7410 was initialized correctly, and **False** if it was not.

Once initialized, you can query the temperature in °C with

```
tempsensor.readTempC()
```

Which will return floating point (decimal + fractional) temperature. You can convert to Fahrenheit by multiplying by 1.8 and adding 32 as you have learned in grade school!

Arduino API

Arduino API (https://adafru.it/DPC)

Python and CircuitPython

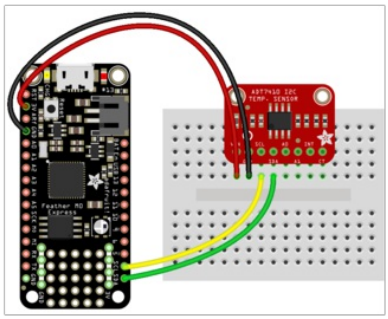
It's easy to use the ADT7410 sensor with Python or CircuitPython, and the Adafruit CircuitPython ADT7410 (https://adafru.it/DPz) module. This module allows you to easily write Python code that reads temperature from the sensor.

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 (https://adafru.it/BSN).

CircuitPython Microcontroller Wiring

First wire up a ADT7410 to your board exactly as shown on the previous page for Arduino.

Make the following connections between the CircuitPython board and the ADT7410:

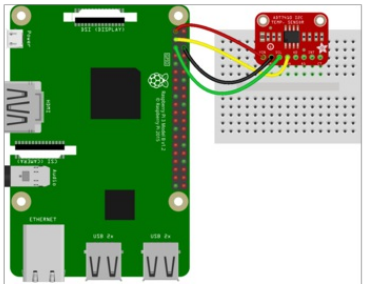


- **Board 3V** to **ADT7410 VIN**
- **Board GND** to **ADT7410 GND**
- **Board SCL** to **ADT7410 SCL**
- **Board SDA** to **ADT7410 SDA**

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 (https://adafru.it/BSN).

Make the following connections between the Pi and the ADT7410:



- **Pi 3V** to **ADT7410 VIN**
- **Pi GND** to **ADT7410 GND**
- **Pi SCL** to **ADT7410 SCL**
- **Pi SDA** to **ADT7410SDA**

CircuitPython Installation of ADT7410 Library

You'll need to install the Adafruit CircuitPython ADT7410 (https://adafru.it/DPA) library on your CircuitPython board.

First make sure you are running the latest version of Adafruit CircuitPython (https://adafru.it/tBa) 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 (https://adafru.it/zdx). For example the Circuit Playground Express guide has a great page on how to install the library bundle (https://adafru.it/C9M) for both express and non-express boards.

Remember for non-express boards like the Trinket M0, Gemma M0, and Feather/Metro M0 basic you'll need to manually install the necessary libraries from the bundle:

- **adafruit_adt7410.mpy**
- **adafruit_bus_device**
- **adafruit_register**

You can also download the**adafruit_adt7410.mpy** from its releases page on Github (https://adafru.it/BfW).

Before continuing make sure your board's lib folder or root filesystem has the **adafruit_adt7410.mpy, adafruit_register,** and **adafruit_bus_device** files and folders copied over.

Next connect to the board's serial REPL (https://adafru.it/Bec)so you are at the CircuitPython >>> prompt.

Python Installation of ADT7410 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 (https://adafru.it/BSN)!

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

- `sudo pip3 install adafruit-circuitpython-adt7410`

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!

Python Usage

To demonstrate the usage of the ADT7410, we'll use the Python REPL.

First, we'll import the following modules:

```
import time
import board
import busio
import adafruit_adt7410
```

Next, we'll initialize the i2c bus and create the ADT object. We'll set it's `high_resolution` property to `True`, to use 16-bit resolution (instead of the default 13-bit).

```
i2c_bus = busio.I2C(board.SCL, board.SDA)
adt = adafruit_adt7410.ADT7410(i2c_bus, address=0x48)
adt.high_resolution = True
```

You can read the temperature using the `.temperature` property (the output will be in degrees Celsius). Try putting your finger on the sensor or holding it against something cold to see the values change.

```
adt.temperature
```

```
>>> import board
>>> import busio
>>> import adafruit_adt7410
>>> i2c_bus = busio.I2C(board.SCL, board.SDA)
>>> adt = adafruit_adt7410.ADT7410(i2c_bus, address=0x48)
>>> adt.high_resolution = True
>>> adt.temperature
22.9141
>>> adt.temperature
22.9063
>>> adt.temperature
22.9063
```

You can convert to Fahrenheit by multiplying by 1.8 and adding 32 as you have learned in grade school:

```
tempC = adt.temperature
tempF = tempC * 1.8 + 32
tempF
```

```
>>> tempC = adt.temperature
>>> tempF = tempC * 1.8 + 32
>>> tempF
73.2875
```

That's all there is to reading the temperature with the ADT7410. Now you can use the ADT7410 temperature sensor to read the temperature in your project!

Full Example Code

```
import time
import board
import busio
import adafruit_adt7410

i2c_bus = busio.I2C(board.SCL, board.SDA)
adt = adafruit_adt7410.ADT7410(i2c_bus, address=0x48)
adt.high_resolution = True

while True:
    print(adt.temperature)
    time.sleep(0.5)
```

Python Docs

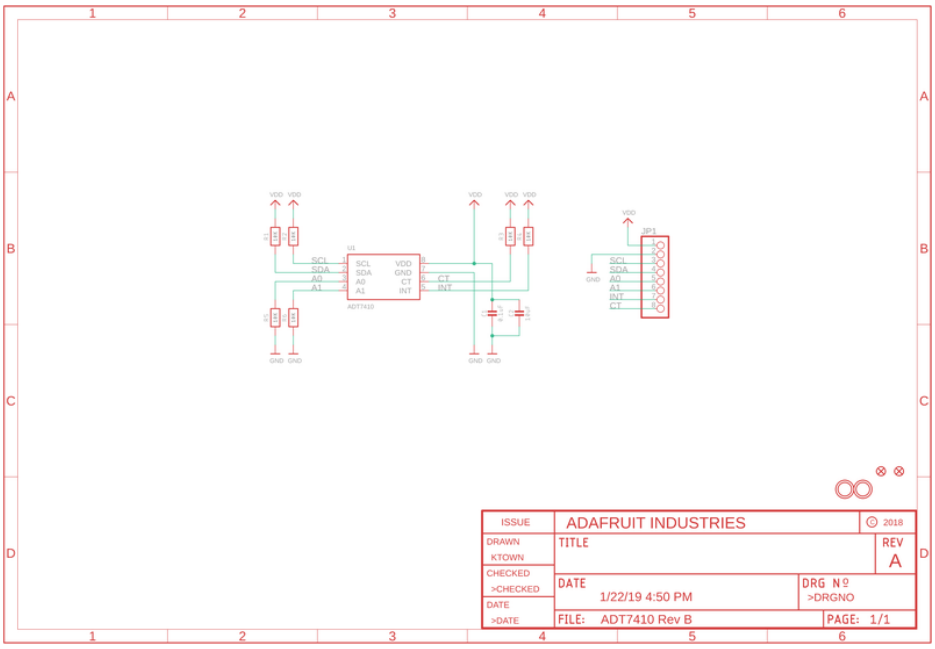
Python Docs (https://adafru.it/DPD)

Downloads

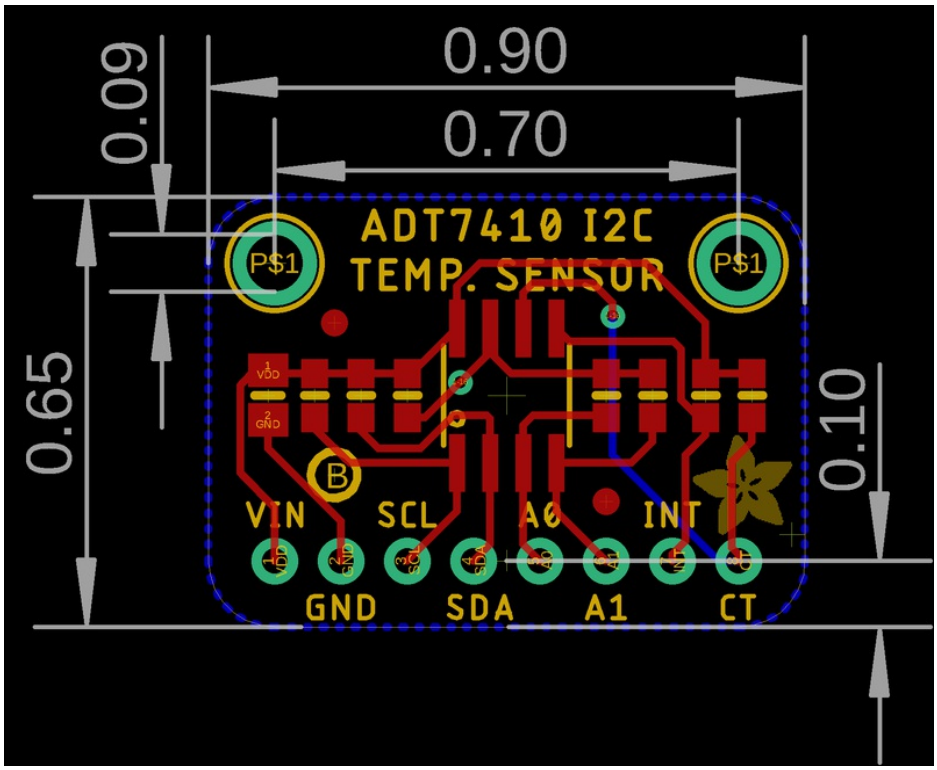
Downloads

- ADT7410 Datasheet (https://adafru.it/DPv)
- Fritzing Object available in the Adafruit Fritzing Library (https://adafru.it/DPw)
- EagleCAD PCB files on GitHub (https://adafru.it/DPx)

Schematic



Fab Print


0.90
0.70
0.09
0.65
0.10
ADT7410 I2C
TEMP. SENSOR
P$1
P$1
VIN
SCL
A0
INT
GND
SDA
A1
CT