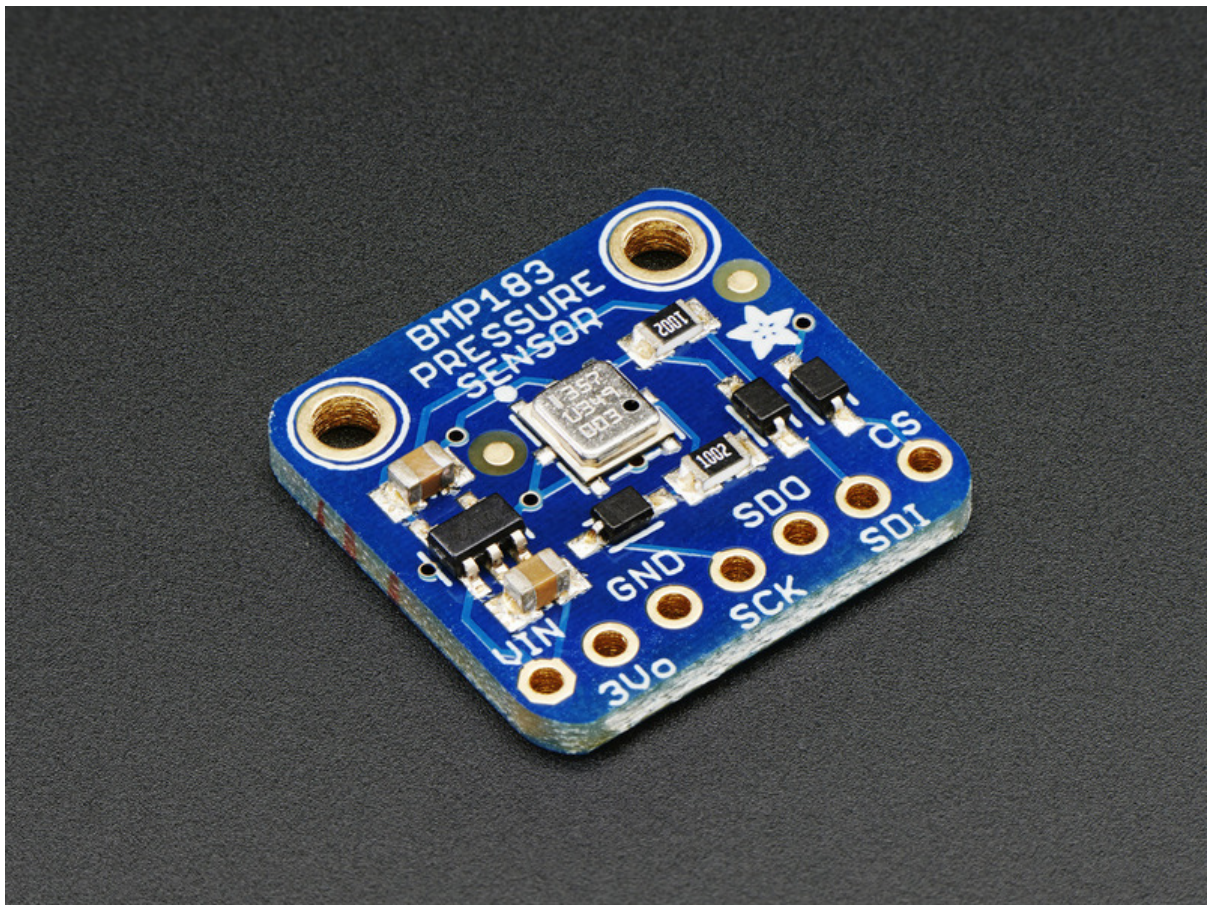




Adafruit BMP183 SPI Barometric Pressure & Altitude Sensor

Created by lady ada



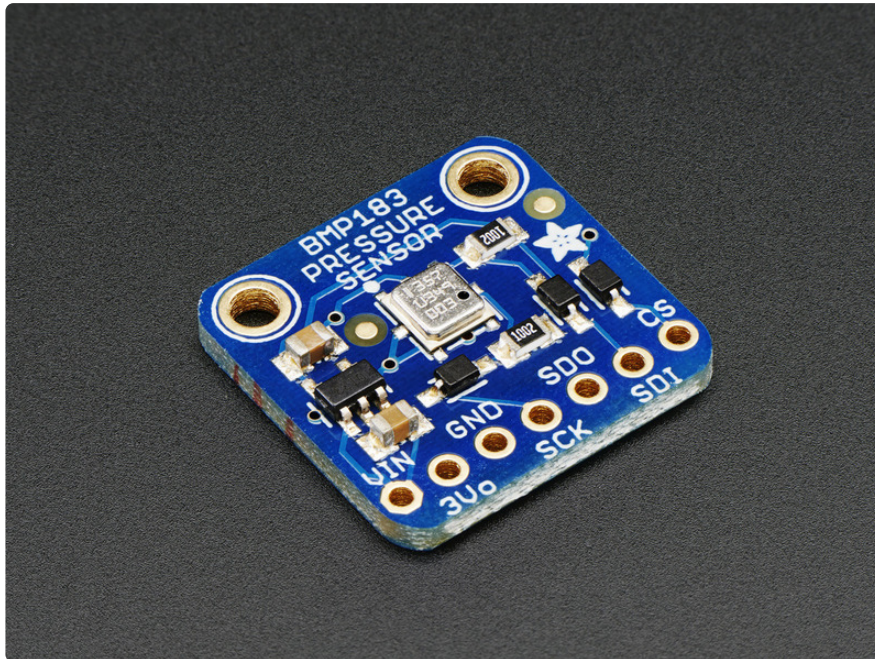
<https://learn.adafruit.com/adafruit-bmp183-spi-barometric-pressure-and-altitude-sensor>

Last updated on 2024-06-03 01:31:28 PM EDT

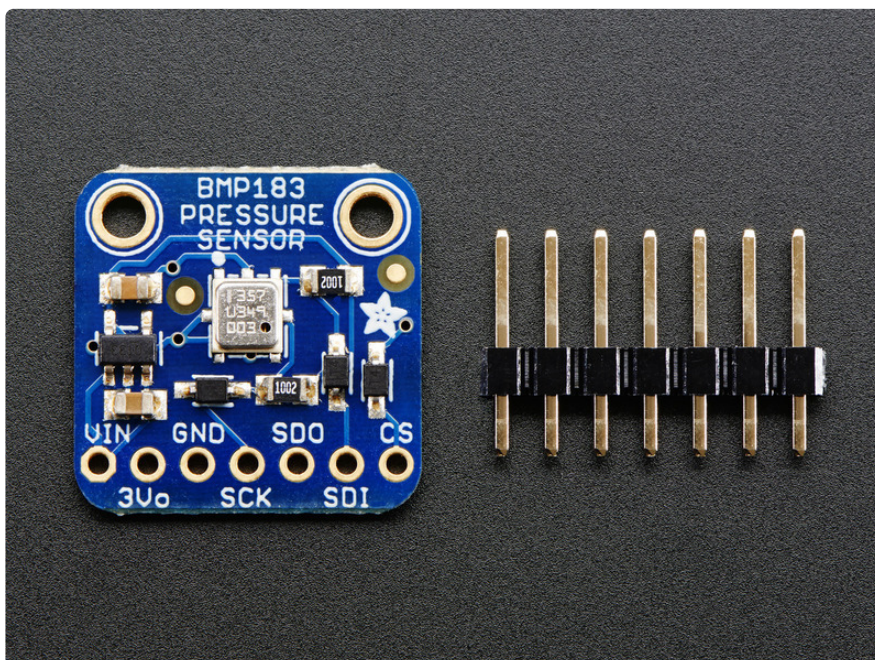
Table of Contents

Overview	3
Pinouts	5
<ul style="list-style-type: none">• Power Pins:• SPI Logic pins:	
Assembly	6
<ul style="list-style-type: none">• Prepare the header strip:• Add the breakout board:• And Solder!	
Wiring and Test	9
<ul style="list-style-type: none">• Download Adafruit_BMP183 library• Load Demo• Library Reference	
Using BMP183 Unified	13
<ul style="list-style-type: none">• Download Adafruit_BMP183_Unified library• Load Demo	
F.A.Q.s	15
Downloads	15
<ul style="list-style-type: none">• Datasheets & Files• Schematic• PCB Fabrication Print	

Overview

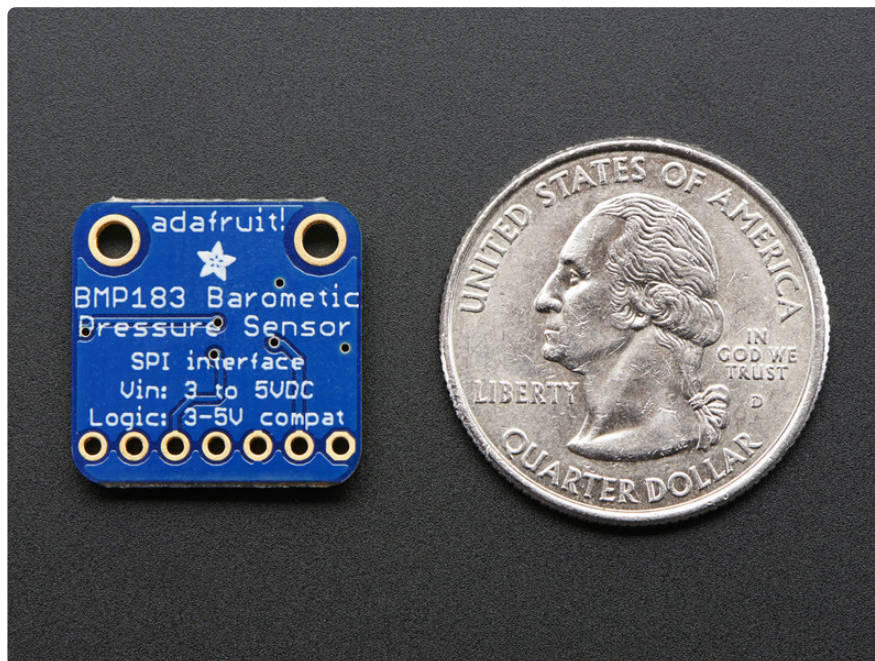


Fans of the BMP085/BMP180 will want to take a look at the new BMP183 - an **SPI** spin on the old familiar classic. This precision sensor from Bosch is the best low-cost sensing solution for measuring barometric pressure and temperature. Because pressure changes with altitude you can also use it as an altimeter!



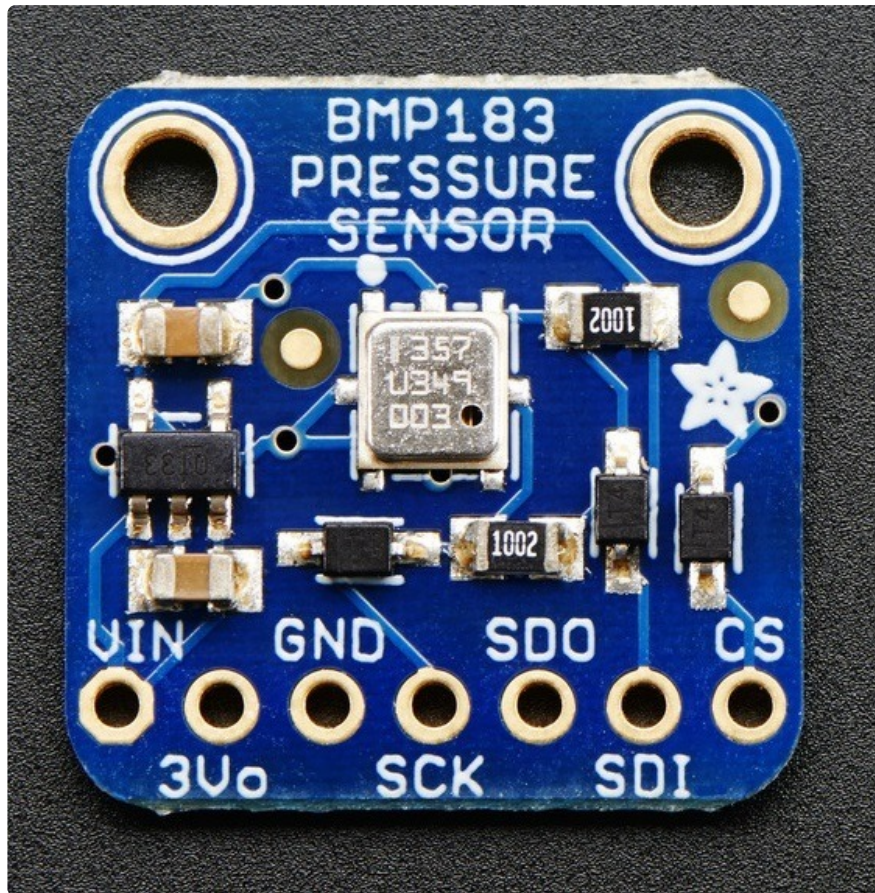
The BMP183 is the next-generation of sensors from Bosch, and is the fraternal twin of the BMP180 - with a low altitude noise of 0.25m and the same fast conversion time. It has the same specifications, but uses SPI instead of I2C. This is great for users where

there is an I2C address collision, they want more than one sensor on a single microcontroller, more flexibility on pin usage, or just prefer the simplicity of SPI.



The sensor is soldered onto a PCB and comes with a 3.3V regulator (so you can use it with 3-5V power), SPI level shifter (so you can use it with 3-5V logic).

Pinouts



Power Pins:

- **Vin** - this is the power pin. Since the chip uses 3 VDC, we have included a voltage regulator on board that will take 3-5VDC and safely convert it down. To power the board, give it the same power as the logic level of your microcontroller - e.g. for a 5V micro like Arduino, use 5V
- **3Vo** - this is the 3.3V output from the voltage regulator, you can grab up to 100mA from this if you like
- **GND** - common ground for power and logic

SPI Logic pins:

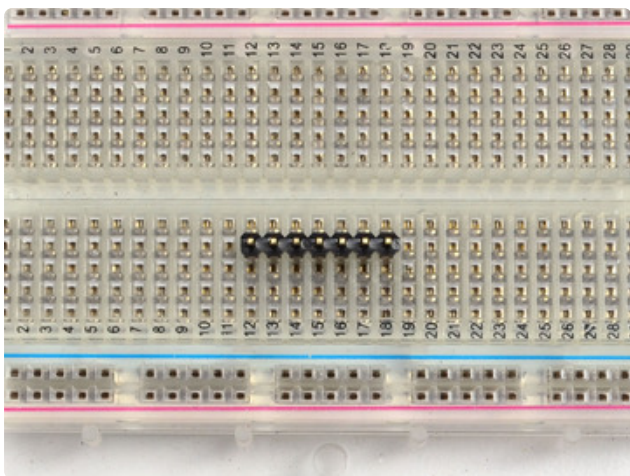
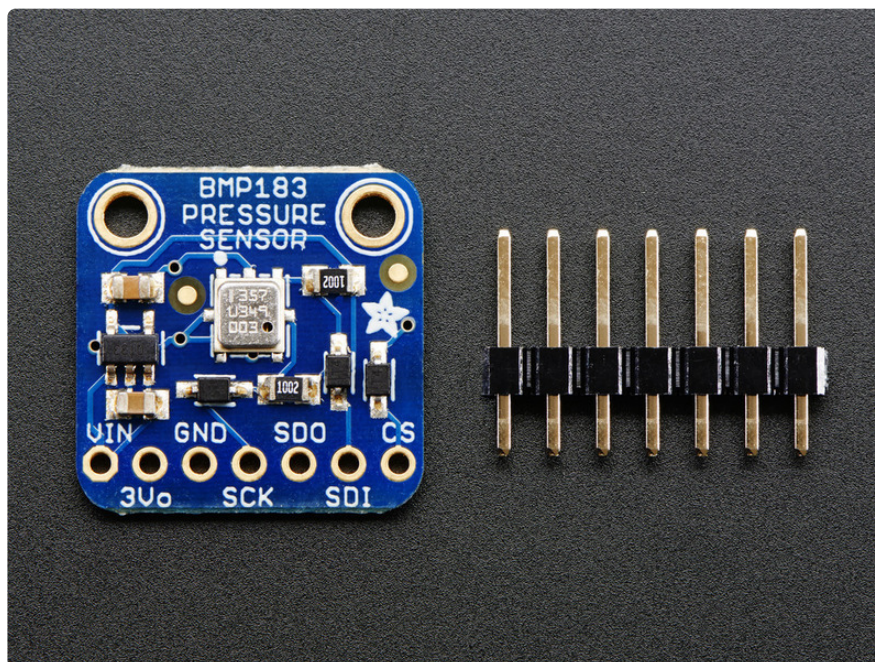
All pins going into the breakout have level shifting circuitry to make them 3-5V logic level safe. Use whatever logic level is on **Vin**!

- **SCK** - This is the **SPI Clock** pin, its an input to the chip

- **SDO** - this is the **S**erial **D**ata **O**ut / **M**icrocontroller **I**n **S**ensor **O**ut pin, for data sent from the BMP183 to your processor
- **SDI** - this is the **S**erial **D**ata **I**n / **M**icrocontroller **O**ut **S**ensor **I**n pin, for data sent from your processor to the BMP183
- **CS** - this is the **C**hip **S**elect pin, drop it low to start an SPI transaction. Its an input to the chip

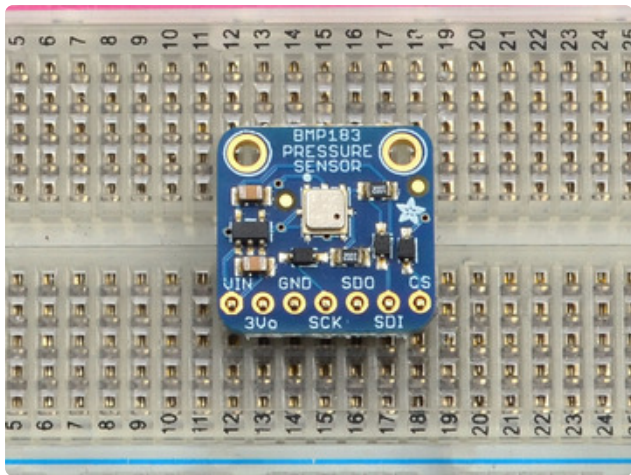
If you want to connect multiple BMP183's to one microcontroller, have them share the SDI, SDO and SCK pins. Then assign each one a unique CS pin.

Assembly



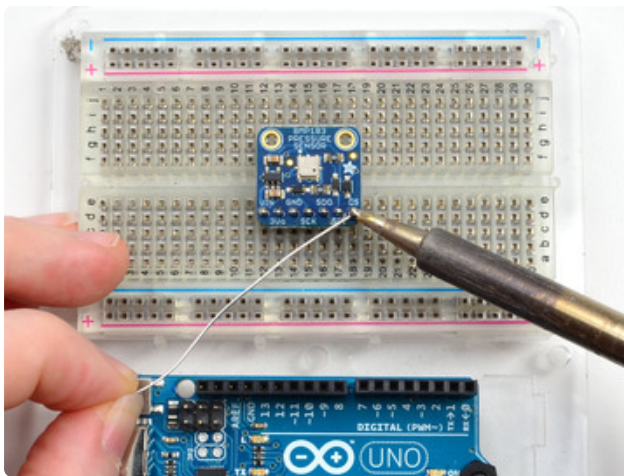
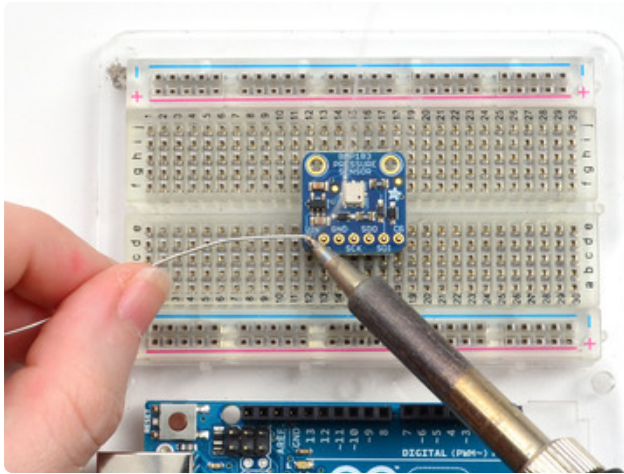
Prepare the header strip:

Cut the strip to length if necessary. It will be easier to solder if you insert it into a breadboard - long pins down



Add the breakout board:

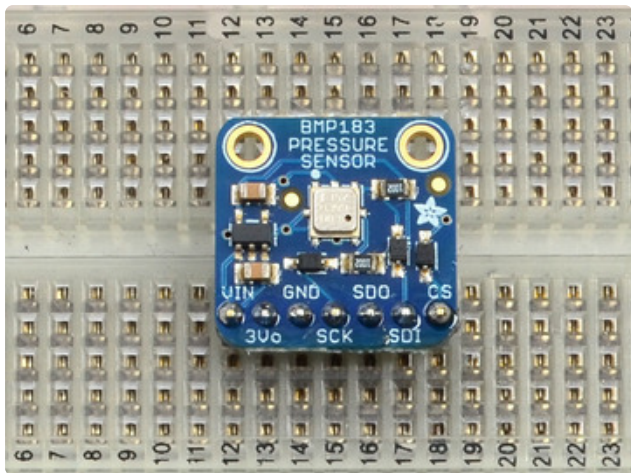
Place the breakout board over the pins so that the short pins poke through the breakout pads



And Solder!

Be sure to solder all pins for reliable electrical contact.

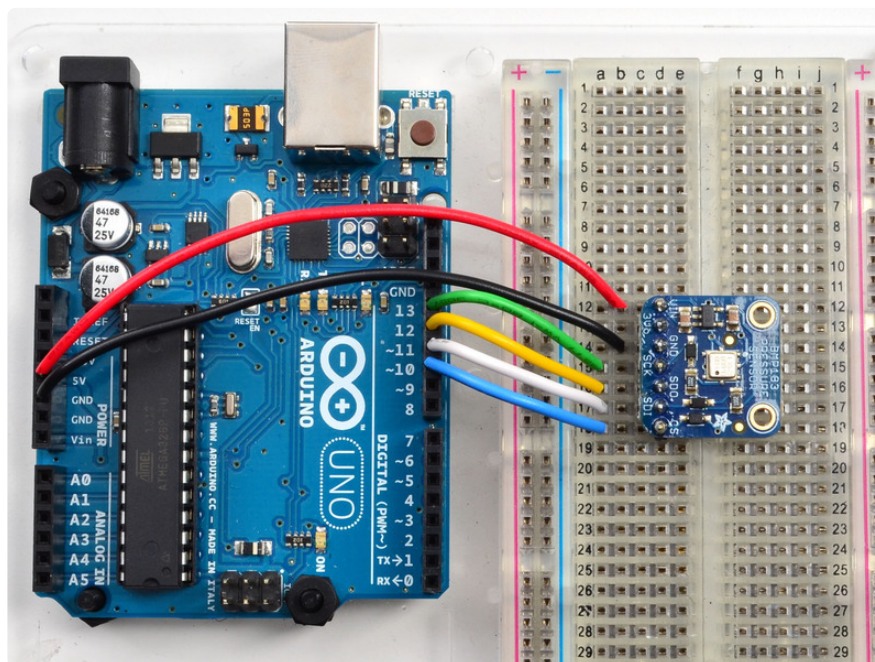
(For tips on soldering, be sure to check out our [Guide to Excellent Soldering \(https://adafruit.it/aTk\)](https://adafruit.it/aTk)).



You're done! Check your solder joints visually and continue onto the next steps

Wiring and Test

You can easily wire this breakout to any microcontroller, we'll be using an Arduino. For another kind of microcontroller, as long as you have 4 available pins it is possible to 'bit-bang SPI'. Check out the library, then port the code.



Since this is a SPI sensor, we can use hardware or 'software' SPI. To make wiring identical on all Arduinos, we'll begin with 'software' SPI. The following pins should be used:

- Connect **Vin** to the power supply, 3V or 5V is fine. Use the same voltage that the microcontroller logic is based off of. For most Arduinos, that is 5V
- Connect **GND** to common power/data ground
- Connect the **SCK** pin to **Digital #13** but any pin can be used later
- Connect the **SDO** pin to **Digital #12** but any pin can be used later
- Connect the **SDI** pin to **Digital #11** but any pin can be used later

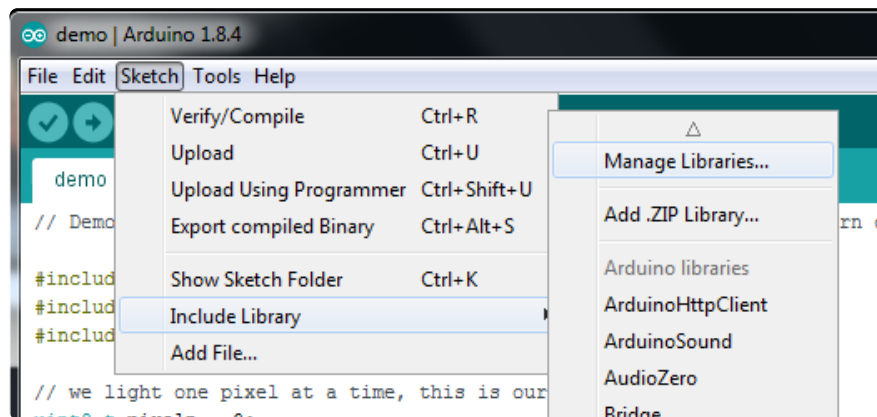
- Connect the **CS** pin **Digital #10** but any pin can be used later

Later on, once we get it working, we can adjust the library to use hardware SPI if you desire, or change the pins to other

Download Adafruit_BMP183 library

To begin reading sensor data, you will need to download the **Adafruit BMP183** library from the Arduino library manager.

Open up the Arduino library manager:



Search for the **Adafruit BMP183** library and install it

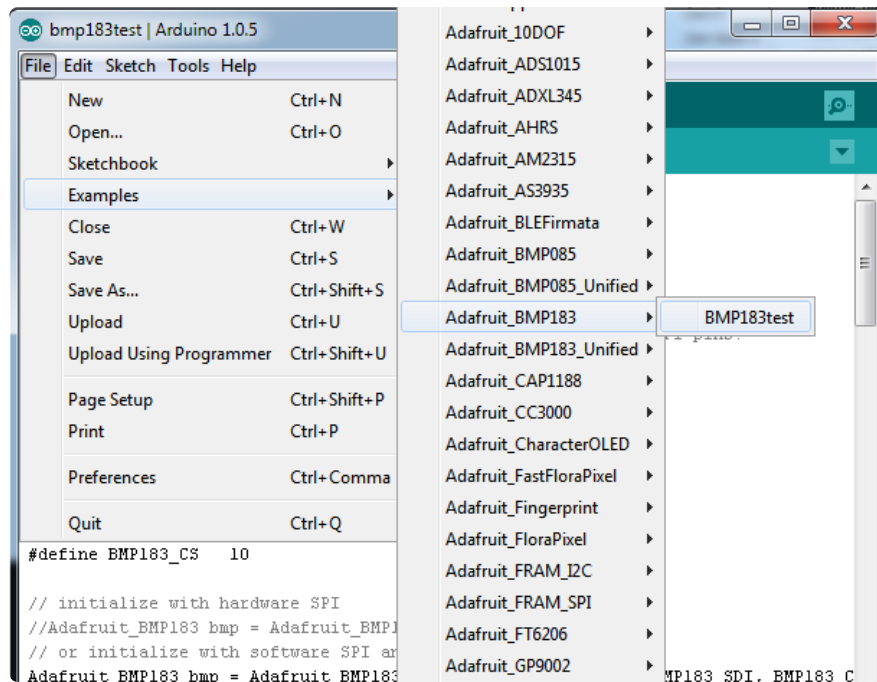


We also have a great tutorial on Arduino library installation at:

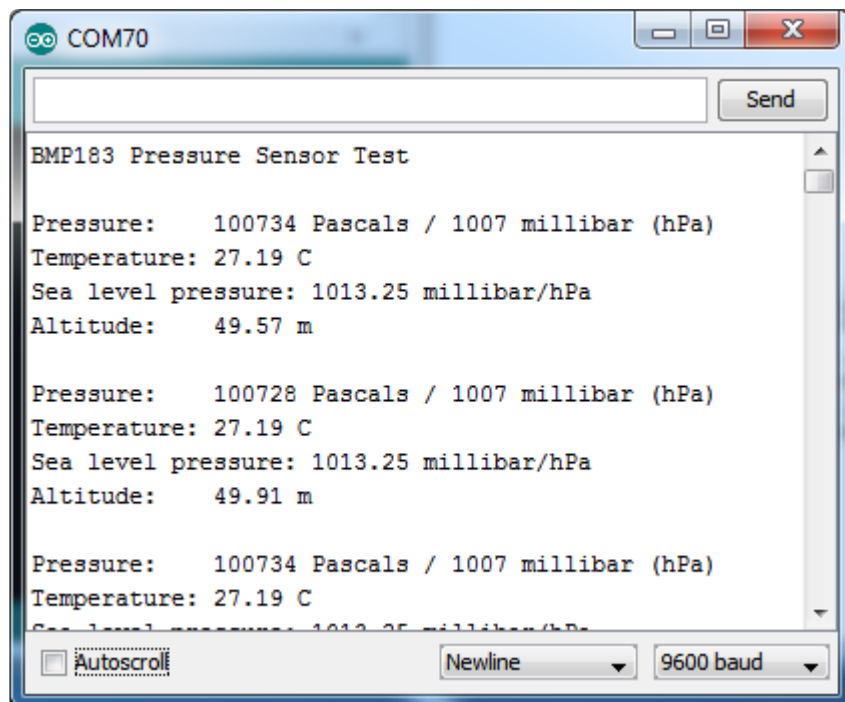
<http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use> (<https://adafru.it/aYM>)

Load Demo

Open up **File->Examples->Adafruit_BMP183->BMP183test** and upload to your Arduino wired up to the sensor



Once uploaded to your Arduino, open up the serial console at 9600 baud speed to see data being printed out



Temperature is calculated in degrees C, you can convert this to F by using the classic $F = C * 9/5 + 32$ equation.

Pressure is returned in the SI units of **Pascals**. 100 Pascals = 1 hPa = 1 millibar. Often times barometric pressure is reported in millibar or inches-mercury. For future reference 1 pascal = 0.000295333727 inches of mercury, or 1 inch Hg = 3386.39 Pascal. So if you take the pascal value of say 100734 and divide by 3389.39 you'll get

29.72 inches-Hg.

You can also calculate Altitude. **However, you can only really do a good accurate job of calculating altitude if you know the hPa pressure at sea level for your location and day!** The sensor is quite precise but if you do not have the data updated for the current day then it can be difficult to get more accurate than 10 meters.

Library Reference

You can start out by creating a BMP183 object with either software SPI (where all four pins can be any I/O) using

```
Adafruit_BMP183 bmp = Adafruit_BMP183(BMP183_CLK, BMP183_SD0, BMP183_SDI, BMP183_CS);
```

Or you can use hardware SPI. With hardware SPI you must use the hardware SPI pins for your Arduino - and each arduino type has different pins! [Check the SPI reference to see what pins to use. \(https://adafru.it/d5h\)](https://adafru.it/d5h)

In this case, you can use any CS pin, but the other three pins are fixed

```
Adafruit_BMP183 bmp = Adafruit_BMP183(BMP183_CS);
```

Once started, you can initialize the sensor with

```
bmp.begin()
```

begin() will return True if the sensor was found, and False if not. If you get a False value back, check your wiring!

Reading temperature and pressure is easy, just call:

```
bmp.getTemperature()  
bmp.getPressure()
```

Temperature is always a floating point, in Centigrade. Pressure is a 32 bit integer with the pressure in Pascals. You may need to convert to a different value to match it with your weather report

It's also possible to turn the BMP183 into an altimeter. If you know the pressure at sea level, the library can calculate the current barometric pressure into altitude

```
bmp.getAltitude(seaLevelPressure)
```


However, you can only really do a good accurate job of calculating altitude if you know the hPa pressure at sea level for your location and day! The sensor is quite precise but if you do not have the data updated for the current day then it can be difficult to get more accurate than 10 meters.

Pass in the current sea level pressure in **hPa** - so the value will be somewhere around ~1000. You can also test with the generic 1013.25 value.

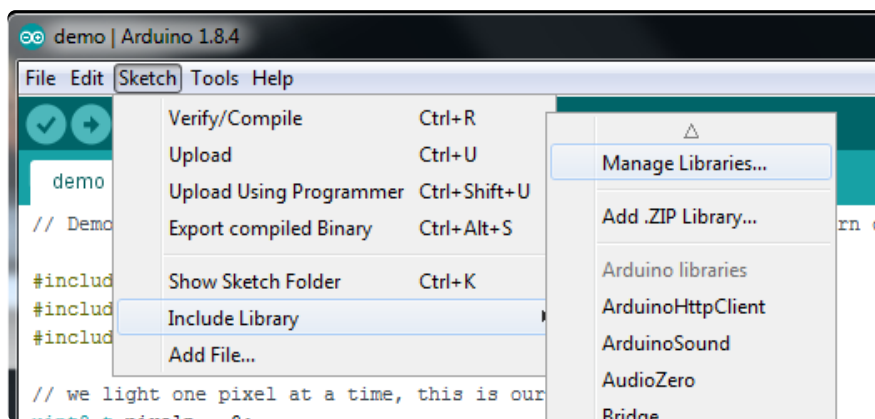
Using BMP183 Unified

The unified sensor library is for slightly more advanced users - it provides a more 'unified' structure for all sensors that Adafruit sells. It mimics the Android sensor platform, and is good for people who are comfortable with data structures.

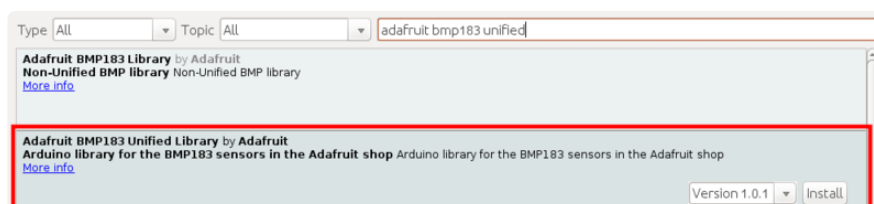
Download Adafruit_BMP183_Unified library

To begin reading sensor data, you will need to download both the **Adafruit BMP183 Unified** library and the **Adafruit Unified Sensor** library from the Arduino library manager.

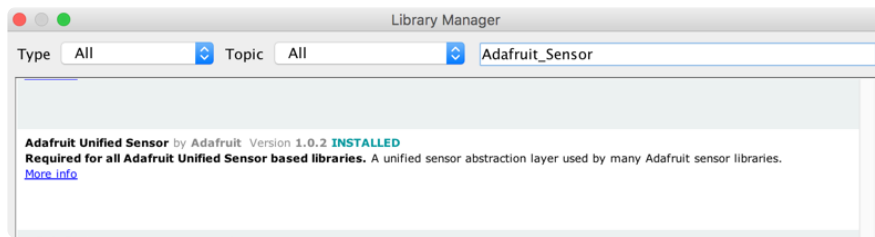
Open up the Arduino library manager:



Search for the **Adafruit BMP183 Unified** library and install it



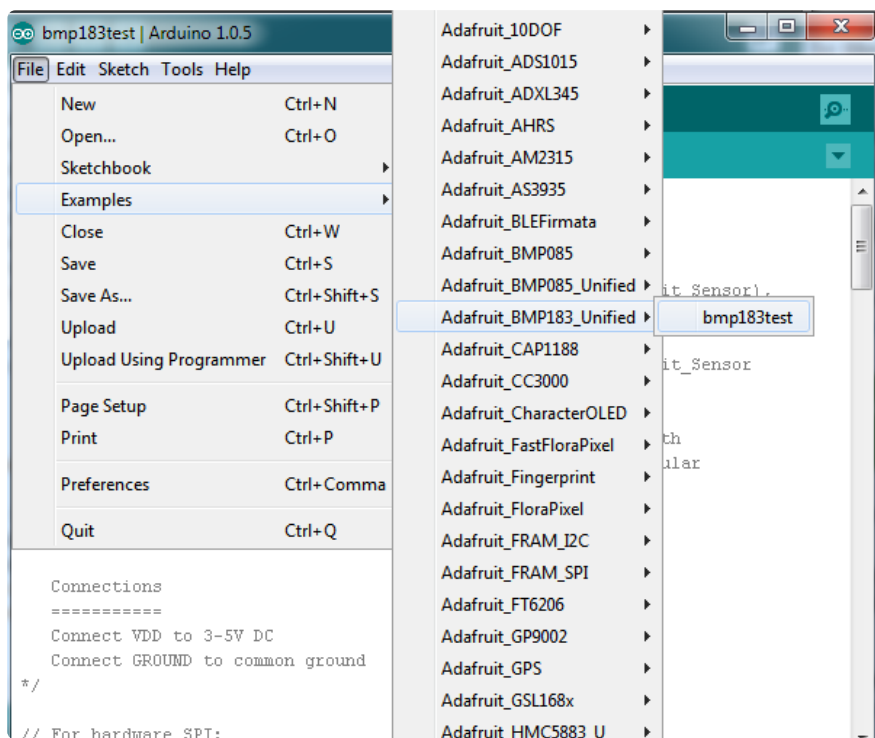
Search for the **Adafruit Sensor** library and install it



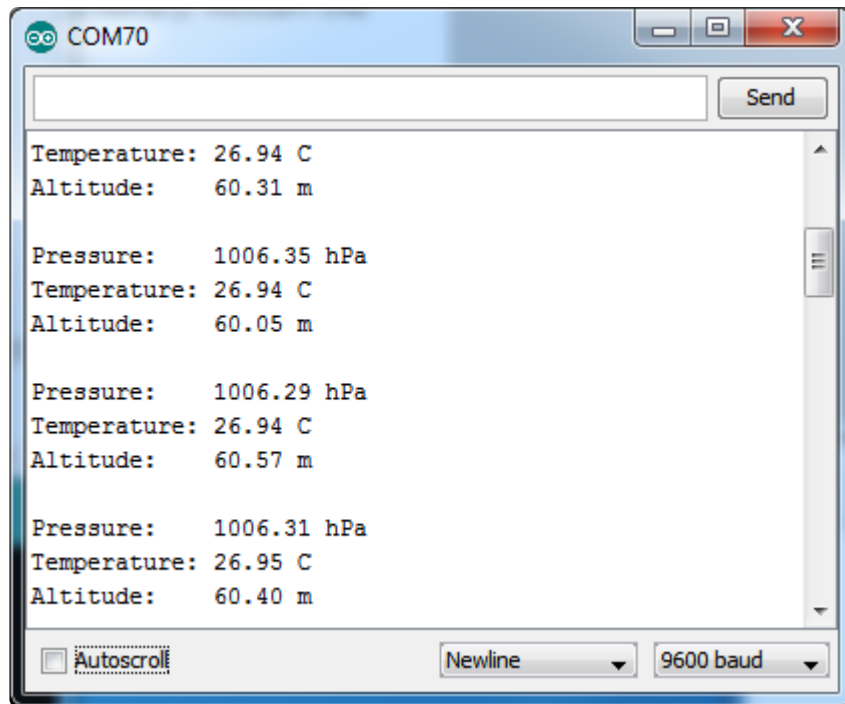
We also have a great tutorial on Arduino library installation at:
<http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use> (<https://adafru.it/aYM>)

Load Demo

Open up **File->Examples->Adafruit_BMP180_Unified->BMP183test** and upload to your Arduino wired up to the sensor



Upload & open up the serial console at 9600 to see the data printed out



F.A.Q.s

How come the altitude calculation is wrong? Is my BMP sensor broken?

No, your sensor is likely just fine. The altitude calculation depends on knowing the barometric pressure at sea level

If you do not set the correct sea level pressure for your location **FOR THE CURRENT DAY** it will not be able to calculate the altitude accurately

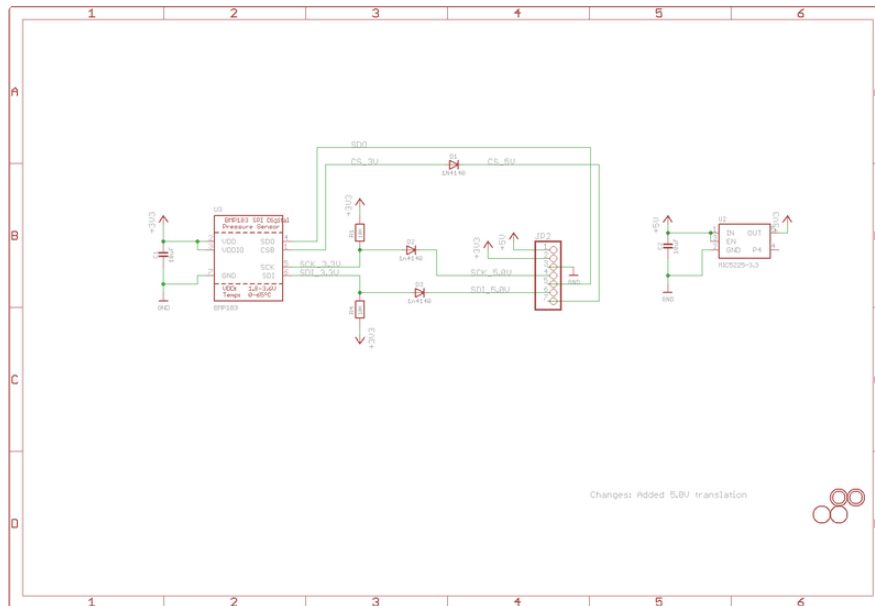
Barometric pressure at sea level changes daily based on the weather!

Downloads

Datasheets & Files

- [Datasheet for the BMP183 sensor used in this breakout \(https://adafru.it/dwD\)](https://adafru.it/dwD)
- [EagleCAD PCB files on GitHub \(https://adafru.it/r5E\)](https://adafru.it/r5E)
- [Fritzing object in Adafruit fritzing library \(https://adafru.it/aP3\)](https://adafru.it/aP3)

Schematic



PCB Fabrication Print

Dimensions in Inches

