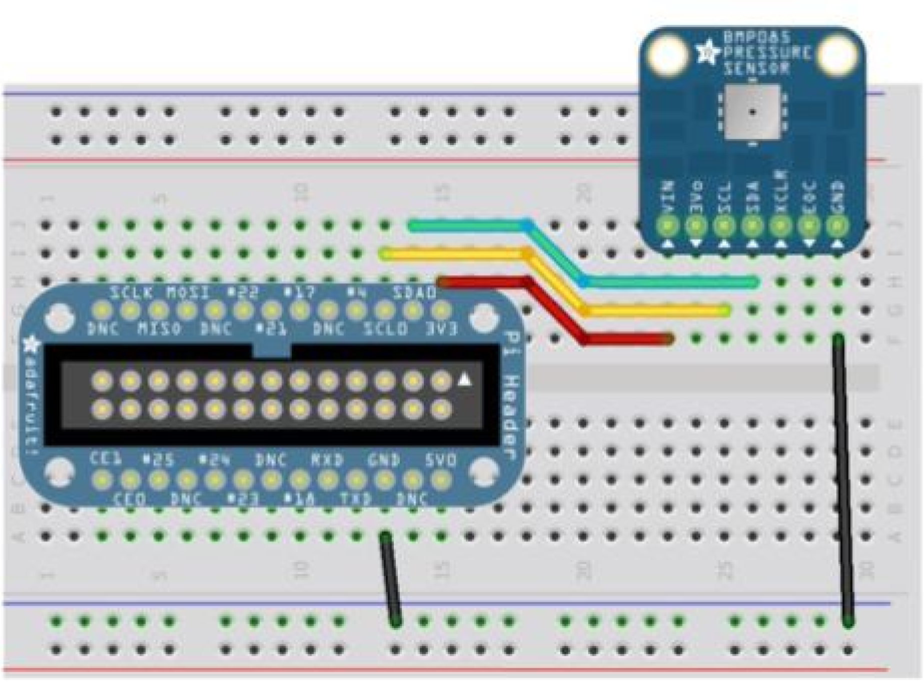
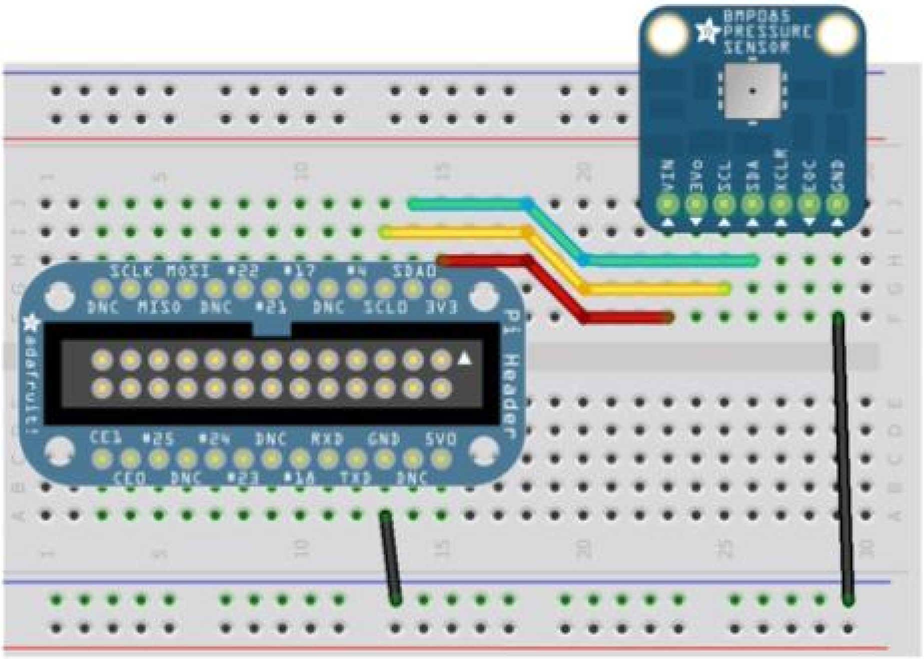
Using the BMP085 with Raspberry Pi



Contents

Overview



The Raspberry Pi includes support for Python, which makes it easy to get access to a lot of low-level hardware and software stacks -- USB, TCP/IP, multiple file systems etc. This is a good thing since it means you don't need to wrap your head around all the obscure details that go along with these complex stacks or the implementation details of various serial buses: you can focus on getting your data off your sensor and into your project as quickly as possible.
Most sensors tend to communicate with other devices based on one of three well-defined mechanisms: **I2C**, **SPI** or good old **analog output**. There are dozens of other serial buses and communication protocols out there (CAN, 1-Wire, etc.), and they all have their strengths and weaknesses, but I2C, SPI and analog cover the overwhelming majority of sensors you're likely to hook up to your development board.

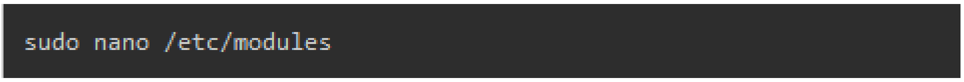
I2C is a particularly useful bus for two main reasons:

- It only requires two shared lines: **SCL** for the clock signal and **SDA** for the bi-direction data transfers.
- Each I2C device uses a unique 7-bit address, meaning you can have more than 120 unique I2C devices sharing the bus, and you can freely communicate with them one at a time on an as-needed basis.

Configuring the Pi for I2C

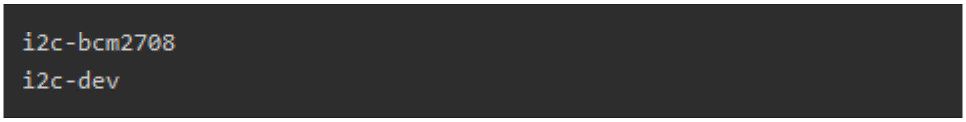
Before you can get started with I2C on the Pi, you'll need to run through a couple quick steps from the console.

If you are using Raspbian, you will need to open LXTerminal and enter the following command:



```
sudo nano /etc/modules
```

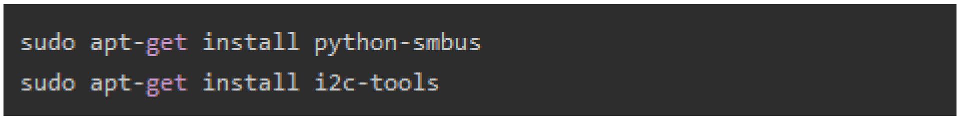
and add these two lines to the end of the file:



```
i2c-bcm2708
i2c-dev
```

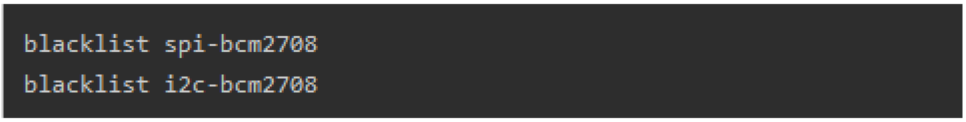
After editing the file, you will need to reboot for the changes to take effect.

Run the following commands in the Terminal to install the i2c-tools utility.



```
sudo apt-get install python-smbus
sudo apt-get install i2c-tools
```

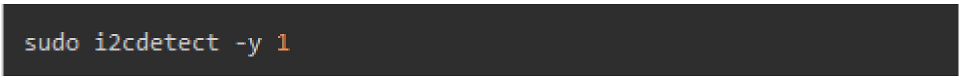
you may also have a file called /etc/modprobe.d/raspi-blacklist.conf you need to edit it and comment out the lines below by putting # in front of them:



```
blacklist spi-bcm2708
blacklist i2c-bcm2708
```

Once this is all done, reboot!

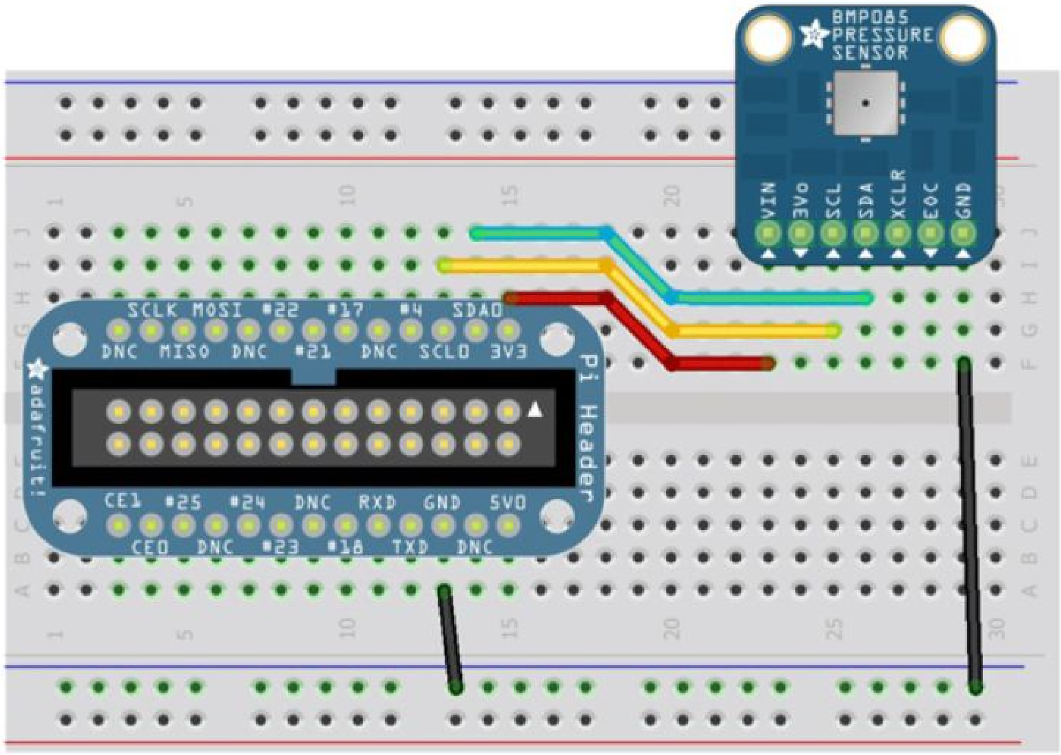
Now when you log in you can type the following command to see all the connected devices (if you are running a 512MB Raspberry Pi Model B)



```
sudo i2cdetect -y 1
```

Hooking Everything Up

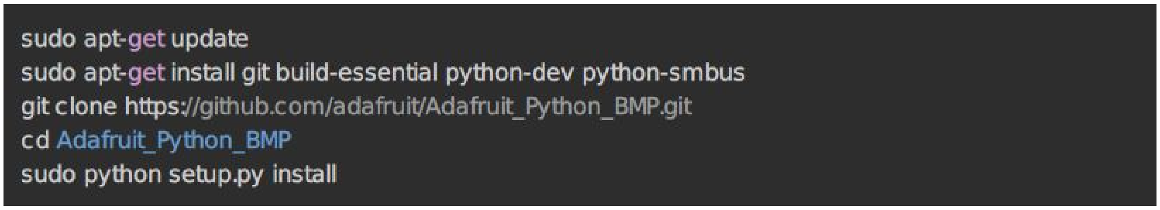
To hook your Adafruit BMP085 Breakout up to the Pi, you can use a Pi Cobbler as seen in the following wiring diagram:



Make sure that you connect the VIN pin on the BMP sensor to 3V3, NOT 5V0! Connecting VIN to 5V supply will cause the board to use 5V logic, which may damage the sensitive 3.3V inputs on the Raspberry Pi.

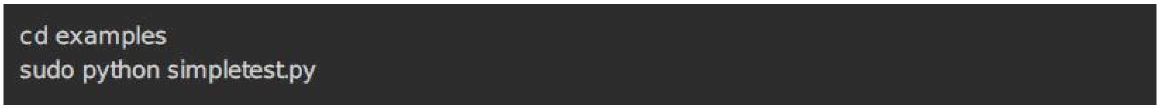
Using the Adafruit BMP Python Library

First make sure your device is powered on and has access to the internet (through a wired or wireless connection). Then connect to your device in a terminal and navigate to a directory where you want to download the library (like /home/pi). Finally execute the following commands to download dependencies and install the library:



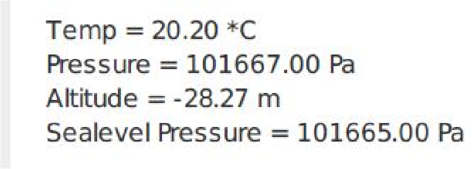
```
sudo apt-get update
sudo apt-get install git build-essential python-dev python-smbus
git clone https://github.com/adafruit/Adafruit_Python_BMP.git
cd Adafruit_Python_BMP
sudo python setup.py install
```

Once the library is installed it will be accessible to any Python script on your device. You can see a few example scripts included in the library source's **examples** folder. Try running the **simpletest.py** example which grabs a single reading from the BMP sensor and displays it by executing:



```
cd examples
sudo python simpletest.py
```

After running the script you should see an output such as:



```
Temp = 20.20 *C
Pressure = 101667.00 Pa
Altitude = -28.27 m
Sealevel Pressure = 101665.00 Pa
```