Bosch BMP085 Breakout Board

Created by Ladyada

Guide Contents

Overview

The BMP085 is a basic sensor that is designed specifically for measuring barometric pressure (it also does temperature measurement on the side to help). It's one of the few sensors that does this measurement, and its fairly low cost so you'll see it used a lot. You may be wondering why someone would want to measure atmospheric pressure, but its actually really useful for two things. One is to measure altitude. As we travel from below sea level to a high mountain, the air pressure *decreases*. That means that if we measure the pressure we can determine our altitude - handy when we don't want the expense or size of a GPS unit. Secondly, atmospheric pressure can be used as a predictor of weather which is why weather-casters often talk about "pressure systems"

> The BMP085 has been discontinued by BOSCH and replaced with the BMP180 which is smaller and less expensive but is identical in terms of wiring and software!
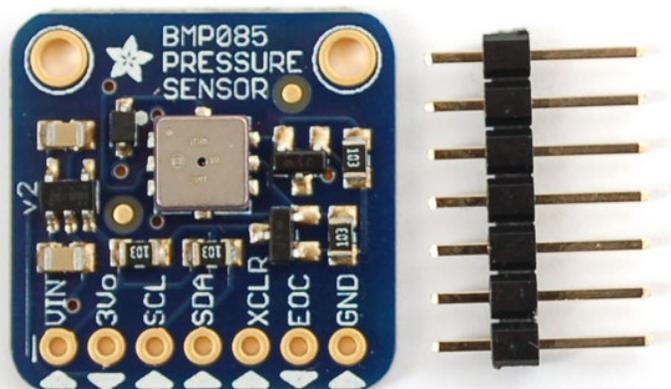
Specifications

- Pressure sensing range: 300-1100 hPa (9000m to -500m above sea level)
- Up to 0.03hPa / 0.25m resolution
- -40 to +85°C operational range, +-2°C temperature accuracy
- 2-pin i2c interface on chip
- V1 of the breakout uses 3.3V power and logic level only
- V2 of the breakout uses 3.3-5V power and logic level for more flexible usage
- Datasheet (http://adafru.it/aKE)

**You can pick up a BMP085 breakout board from the Adafruit shop! (http://adafru.it/391)**

Wiring the BMP085

Since the BMP085 is a i2c sensor, its very easy to wire up. We'll be using an Arduino as an example but any microcontroller with i2c can be used. To start using with a solderless breadboard, we need to solder the header pins onto the breakout board.

We suggest plugging the header into a breadboard so the long pins are in the breadboard, and then laying the BMP085 breakout board on top. The photos show a v1 BMP085 sensor but this part is identical for both versions!
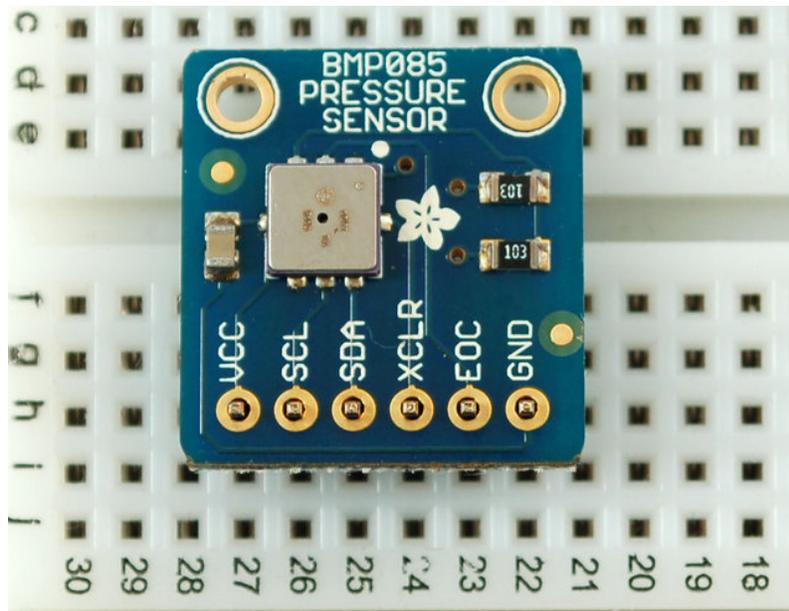
Then solder all of the pins!
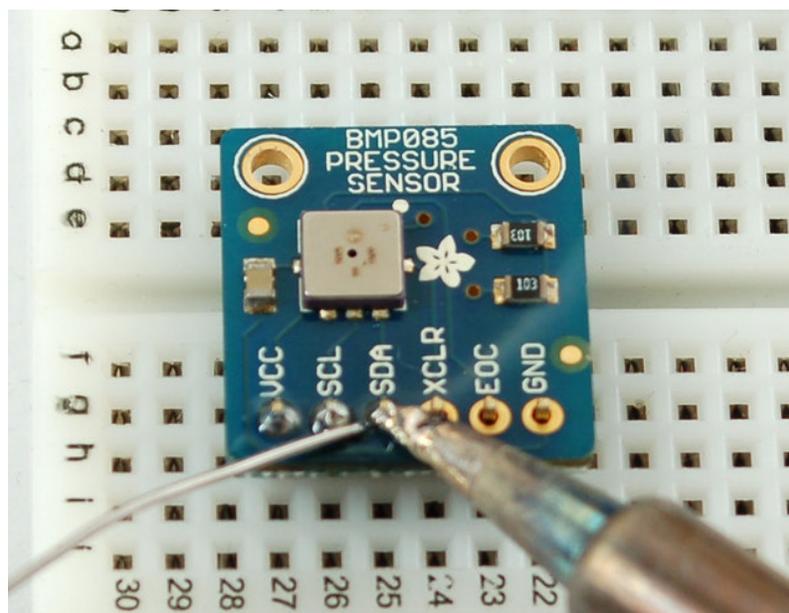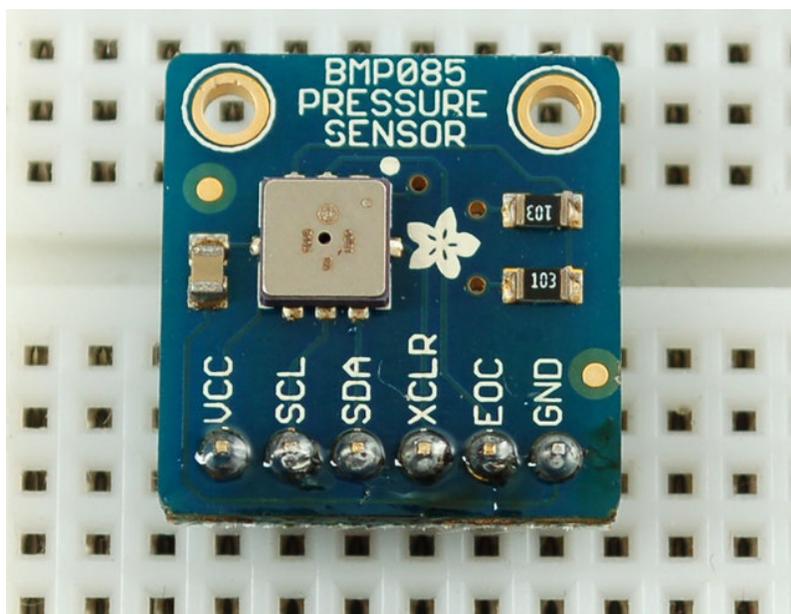
That's it! Now we can wire the board up to the microcontroller.
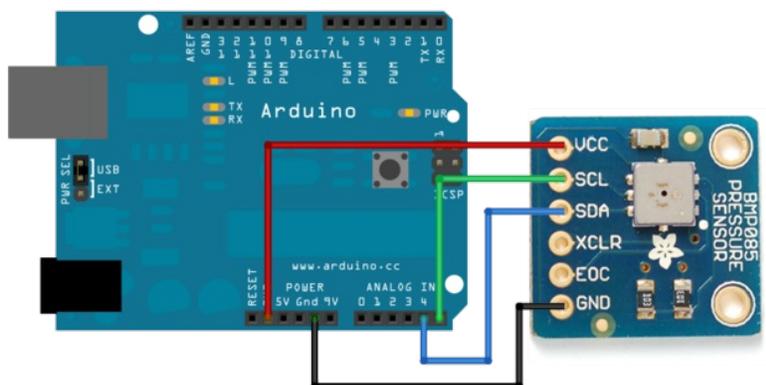
Connect the **VCC** pin to a **3.3V** power source. The V1 of the sensor breakout cannot be used with anything higher than 3.3V so don't use a 5V supply! V2 of the sensor board has a 3.3V regulator so you can connect it to either 3.3V or 5V if you do not have 3V available.

Connect **GND** to the ground pin.

Connect the **i2c SCL clock** pin to your i2c clock pin. On the classic Arduino Uno/Duemilanove/Diecimila/etc this is **Analog pin #5**

Connect the **i2c SDA data** pin to your i2c data pin. On the classic Arduino Uno/Duemilanove/Diecimila/etc this is **Analog pin #4**

Unfortunately, the i2c lines on most microcontrollers are fixed so you're going to have to stick with those pins.

You don't need to connect the **XCLR** (reset) or **EOC** (end-of-conversion) pins. If you need to speed up your conversion time, you can use the EOC as a indicator - in our code we just hang out and wait the maximum time possible.

Using the BMP085 (API v2)

This page is based on the new v2 of the BMP085 driver, which uses Adafruit's new Unified Sensor Driver (http://adafru.it/cgk). The driver provides better support for altitude calculations, and makes it easy to switch between the BMP085 and any other supported pressure sensor in your projects.

If you haven't already done so, you'll need to install the Adafrut_Sensor library (http://adafru.it/aZm) on your system as well, since Adafruit_BMP085 relies on this library to generate the sensor data in a universal manner.

Using the BMP085

To use this sensor and calculate the altitude and barometric pressure, there's a lot of very hairy and unpleasant math. You can check out the math in the datasheet but really, its not intuitive or educational - its just how the sensor works. So we took care of all the icky math and wrapped it up into a nice Arduino library.

You can find the Arduino library repository on github (http://adafru.it/aZq) To install it, click this button to download the compressed ZIP file then install it. This guide (http://adafru.it/aYM) will help you with the install process if you have never installed an Arduino library.

Download the
Adafruit_BMP085
Arduino Library
(API v2)

http://adafru.it/cFb

Restart the IDE

Now you can run this first example sketch

```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP085.h>

Adafruit_BMP085 bmp = Adafruit_BMP085(10085);

void setup(void)
{
  Serial.begin(9600);
  Serial.println("Pressure Sensor Test"); Serial.println("");

  /* Initialise the sensor */
  if(!bmp.begin())
  {
    /* There was a problem detecting the BMP085 ... check your connections */
```

```
    Serial.print("Ooops, no BMP085 detected ... Check your wiring or I2C ADDR!");
    while(1);
  }
}

void loop(void)
{
  /* Get a new sensor event */
  sensors_event_t event;
  bmp.getEvent(&event);

  /* Display the results (barometric pressure is measure in hPa) */
  if (event.pressure)
  {
    /* Display atmospheric pressure in hPa */
    Serial.print("Pressure: "); Serial.print(event.pressure); Serial.println(" hPa");
  }
  else
  {
    Serial.println("Sensor error");
  }
  delay(250);
}
```

Then open up the serial monitor at 9600 baud. The sketch will continuously print out the pressure in **hPa** (hectoPascals). You can test that the sensor is measuring variations in pressure by placing your fingertip over the open port hole in the top of the sensor. The pressure will increase as you can see here:
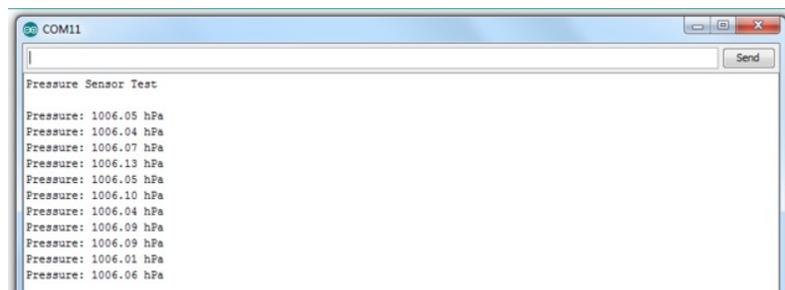
Altitude Measurements

Since we know that pressure drops as we gain altitude (that's why air is so thin on mountain-tops) we can compute the current altitude knowing the pressure and temperature. Again, there's a bit of hairy math involved, you can read about the calculations on wikipedia (where this graph is from) (http://adafru.it/aKG).

Atmospheric Pressure vs. Altitude

Atmospheric Pressure (kPa)

```
110
100
 90
 80
 70
 60
 50
 40
 30
 20    Computed for 15 deg. C and 0% humidity
 10
                                              Altitude (m)
        2000    4000    6000    8000   10000
-10
```
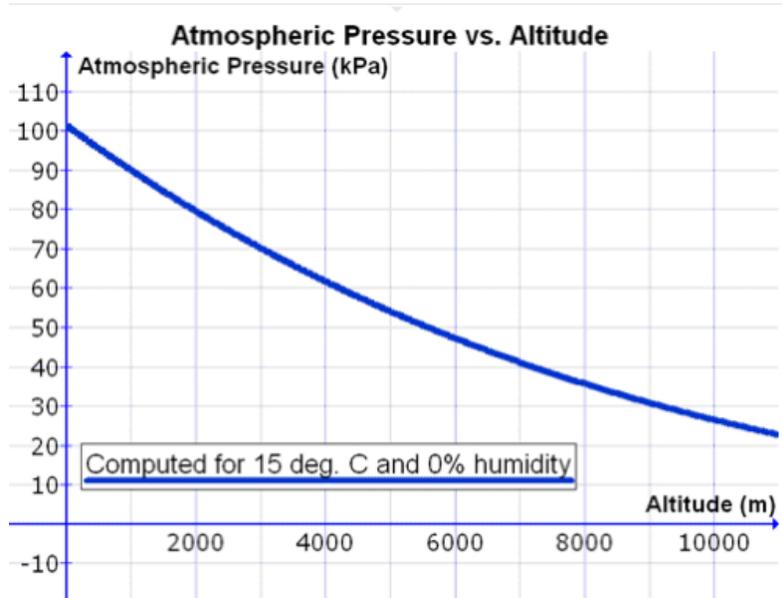
With the Arduino library, we take care of that for you! Simply **update the 'void loop()' function above with the code below to get the altitude based on the pressure and temperature:**

```
void loop(void)
{
  /* Get a new sensor event */
  sensors_event_t event;
  bmp.getEvent(&event);

  /* Display the results (barometric pressure is measure in hPa) */
  if (event.pressure)
  {
    /* Display atmospheric pressue in hPa */
    Serial.print("Pressure:    ");
    Serial.print(event.pressure);
    Serial.println(" hPa");

    /* Calculating altitude with reasonable accuracy requires pressure    *
     * sea level pressure for your position at the moment the data is      *
     * converted, as well as the ambient temperature in degress           *
     * celcius.  If you don't have these values, a 'generic' value of      *
     * 1013.25 hPa can be used (defined as SENSORS_PRESSURE_SEALEVELHPA    *
     * in sensors.h), but this isn't ideal and will give variable          *
     * results from one day to the next.                                   *
     *                                                                      *
     * You can usually find the current SLP value by looking at weather    *
     * websites or from environmental information centers near any major  *
     * airport.                                                            *
     *                                                                      *
     * For example, for Paris, France you can check the current mean       *
     * pressure and sea level at: http://bit.ly/16Au8ol                */

    /* First we get the current temperature from the BMP085 */
    float temperature;
    bmp.getTemperature(&temperature);
```

```
      Serial.print("Temperature: ");
      Serial.print(temperature);
      Serial.println(" C");

      /* Then convert the atmospheric pressure, SLP and temp to altitude    */
      /* Update this next line with the current SLP for better results       */
      float seaLevelPressure = SENSORS_PRESSURE_SEALEVELHPA;
      Serial.print("Altitude:    ");
      Serial.print(bmp.pressureToAltitude(seaLevelPressure,
                        event.pressure,
                        temperature));
    Serial.println(" m");
    Serial.println("");
  }
  else
  {
    Serial.println("Sensor error");
  }
  delay(1000);
}
```

Run the sketch to see the calculated altitude.
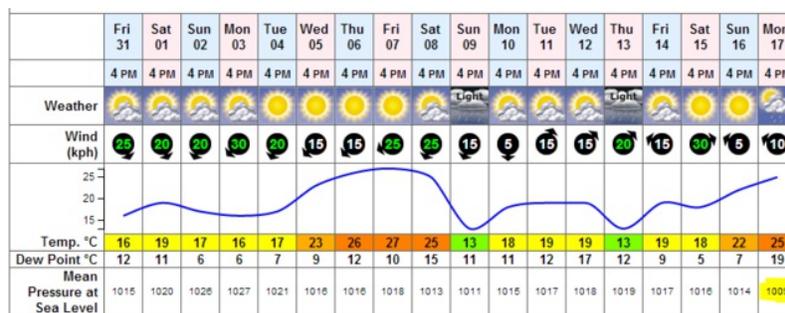
```
Pressure:     1006.02 hPa
Temperature: 27.74 C
Altitude:      63.10 m
```

The data above is reasonably close to what I'd expect at my location, but we can improve the accuracy by changing the reference sea level pressure, which will change depending on the weather conditions. **Every 1 hPa that we are off on the sea level pressure equals about 8.5 m of error in the altitude calculations!**

Many weather sites, (particularly near major airports) will provide pressure readings. If you happened to be near Paris, France, for example, you might look up the current air pressure at Charles de Gaulle airport (http://adafru.it/cgl), which we can see is 1009 hPa (a meaningful difference from the generoc 1013.25 hPa value we are plugging in via the **SENSORS_PRESSURE_SEALEVELHPA** macro):

	Fri 31	Sat 01	Sun 02	Mon 03	Tue 04	Wed 05	Thu 06	Fri 07	Sat 08	Sun 09	Mon 10	Tue 11	Wed 12	Thu 13	Fri 14	Sat 15	Sun 16	Mon 17
	4 PM	4 PM	4 PM	4 PM	4 PM	4 PM	4 PM	4 PM	4 PM	4 PM	4 PM	4 PM	4 PM	4 PM	4 PM	4 PM	4 PM	4 PM
Weather																		
Wind (kph)	25	20	20	30	20	15	15	25	25	15	5	15	15	20	15	30	5	10
Temp. °C	16	19	17	16	17	23	26	27	25	13	18	19	19	13	19	18	22	25
Dew Point °C	12	11	6	6	7	9	12	10	15	11	11	12	17	12	9	5	7	19
Mean Pressure at Sea Level	1015	1020	1026	1027	1021	1016	1016	1018	1013	1011	1015	1017	1018	1019	1017	1016	1014	1009

Updating the following line to 1009 will give us a more accurate altitude:

```
float seaLevelPressure = 1009;
```

> If you don't know the current mean pressure at sea level for your current weather conditions and location, SENSORS_PRESSURE_SEALEVELHPA is still a safe starting point:

This now gives us the following results, which shows that calibrating for your local conditions is often worthwhile when working with low altitudes!

```
Pressure:     1006.05 hPa
Temperature:  28.41 C
Altitude:     25.85 m
```

Just be careful looking for local mean pressure at sea level values, since the functions in the driver are expecting hPa units, not one of the dozens of other values you may encounter, but you should be able to convert anything you find to hPa (http://adafru.it/aKH) which is a standard SI unit.

Using the BMP085 (API v1)

To use this sensor and calculate the altitude and barometric pressure, there's a lot of very hairy and unpleasant math. You can check out the math in the datasheet but really, its not intuitive or educational - its just how the sensor works. So we took care of all the icky math and wrapped it up into a nice Arduino library.

You can find the Arduino library repository on github (http://adafru.it/aKl) To install it, click this button to download the compressed ZIP file then install it. This guide (http://adafru.it/aYM) will help you with the install process if you have never installed an Arduino library.

Download the Adafruit_BMP085 Arduino Library (API v1)

http://adafru.it/cFc

Restart the IDE

Now you can run this first example sketch

```
#include "Wire.h"
#include "Adafruit_BMP085.h"

Adafruit_BMP085 bmp;

void setup() {
  Serial.begin(9600);
  bmp.begin();
}

void loop() {
    Serial.print("Temperature = ");
    Serial.print(bmp.readTemperature());
    Serial.println(" *C");

    Serial.print("Pressure = ");
    Serial.print(bmp.readPressure());
    Serial.println(" Pa");

    Serial.println();
    delay(500);
}
```
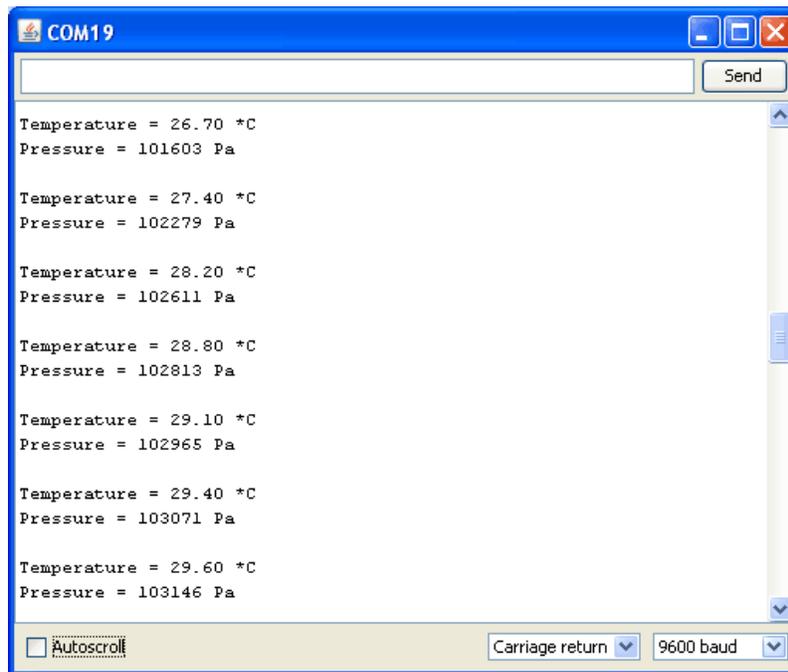
Then open up the serial monitor at 9600 baud. The sketch will continuously print out the temperature in **°C** and pressure in **Pa** (Pascals). You can test that the sensor is measuring variations in temperature and pressure by placing your fingertip over the open port hole in the top of the sensor. The temperature and pressure will increase as you can see here:

```
Temperature = 26.70 *C
Pressure = 101603 Pa

Temperature = 27.40 *C
Pressure = 102279 Pa

Temperature = 28.20 *C
Pressure = 102611 Pa

Temperature = 28.80 *C
Pressure = 102813 Pa

Temperature = 29.10 *C
Pressure = 102965 Pa

Temperature = 29.40 *C
Pressure = 103071 Pa

Temperature = 29.60 *C
Pressure = 103146 Pa
```

Altitude Measurements

Since we know that pressure drops as we gain altitude (that's why air is so thin on mountain-tops) we can compute the current altitude knowing the pressure and temperature. Again, there's a bit of hairy math involved, you can read about the calculations on wikipedia (where this graph is from) (http://adafru.it/aKG).

Atmospheric Pressure vs. Altitude

Computed for 15 deg. C and 0% humidity
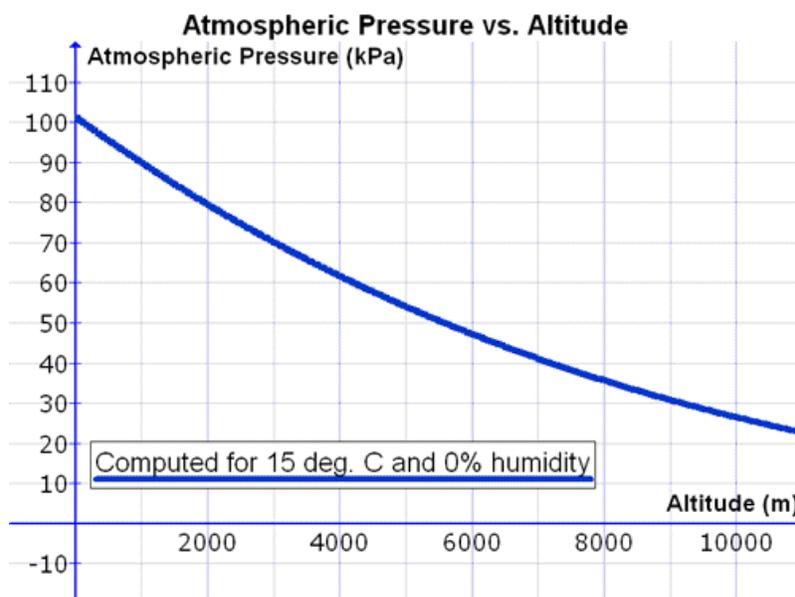
With the Arduino library, we take care of that for you! Simply run this sketch which will return the

current altitude based on the pressure.

```
#include "Wire.h"
#include "Adafruit_BMP085.h"

Adafruit_BMP085 bmp;

void setup() {
  Serial.begin(9600);
  bmp.begin();
}

void loop() {
    Serial.print("Temperature = ");
    Serial.print(bmp.readTemperature());
    Serial.println(" *C");

    Serial.print("Pressure = ");
    Serial.print(bmp.readPressure());
    Serial.println(" Pa");

    // Calculate altitude assuming 'standard' barometric
    // pressure of 1013.25 millibar = 101325 Pascal
    Serial.print("Altitude = ");
    Serial.print(bmp.readAltitude());
    Serial.println(" meters");

    Serial.println();
    delay(500);
}
```
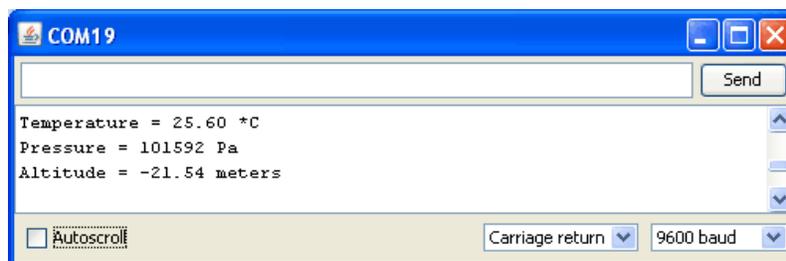
Run the sketch to see the calculated altitude.

For example, according to the sensor we are 21.5m below sea level. Only problem is, I know for a fact that our current location is not below sea level! So what's wrong with the sensor? Turns out the sensor is just fine. The problem is that the pressure at sea level changes with the weather. So we need to 'normalize' the sensor, and let it know what the sea-level pressure is. You can look up the current sea level pressure on any weather site.

Right Now

Cloudy

82°F

Feels Like: 82°
Get FREE weather on your desktop

Past 24-hr:
Precip: 0 in
Snow: 0 in

Wind:
From SW at 11mph

Through 3am: Clear with temperatures
steady near 74F. Winds SW at 5 to 10
mph.

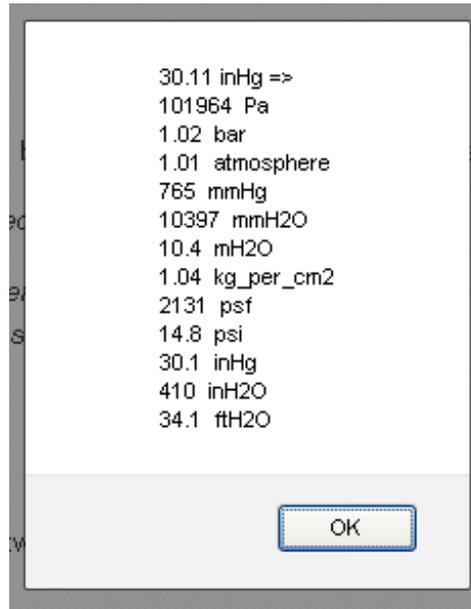Humidity: 45%

Dew Point: 59

Pressure: 30.11 in →

Visibility: 7.0 mi

UV Index: 0 - Low

Time Until Sunrise: 8 hrs 43 min

Unfortunately there are half-dozen different units of pressure. here we see it in inches, that's technically "Mercury Inches" or "Hg Inches We need it in Pascals, so we'll convert it! (http://adafru.it/aKH)

```
30.11 inHg =>
101964 Pa
1.02 bar
1.01 atmosphere
765 mmHg
10397 mmH2O
10.4 mH2O
1.04 kg_per_cm2
2131 psf
14.8 psi
30.1 inHg
410 inH2O
34.1 ftH2O
```

OK so that's 101,964 Pascals. Open up the **Examples->BMP085test** example from the Arduino IDE menubar and edit the line where you pass in the 'corrected' altitude.

```
// you can get a more precise measurement of altitude
// if you know the current sea level pressure which will
// vary with weather and such. If it is 1015 millibars
// that is equal to 101500 Pascals.
Serial.print("Real altitude = ");
Serial.print(bmp.readAltitude(101964));
Serial.println(" meters");
```

Now it will print out the correct altitude! 30 meters which is a lot better.

```
Temperature = 25.50 *C
Pressure = 101612 Pa
Altitude = -23.29 meters
Real altitude = 29.58 meters
```
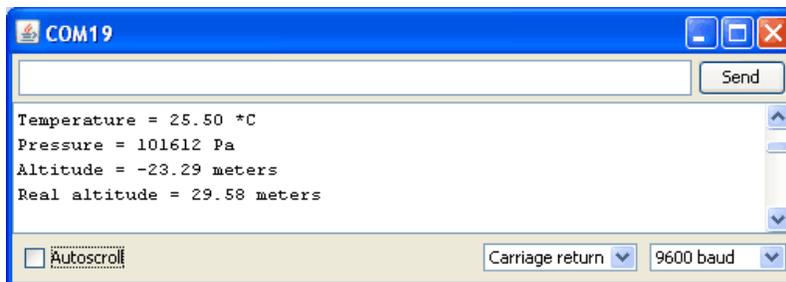
Downloads

Unified BMP085 Driver Repository (New)

Github repository for the 'Unified' sensor driver, inlcuding example code (http://adafru.it/aZq)

For new projects, it's highly recommended to use the new Unified BMP085 Driver, which makes it easier to switch between other pressure sensors, and work with standard SI units of measure.

To download, click the DOWNLOAD button in the top right corner, renamed the compressed folder Adafruit_BMP085. Check that the Adafruit_BMP085 folder contains Adafruit_BMP085.cpp and Adafruit_BMP085.h.

Place the Adafruit_BMP085 library folder in your /libraries folder. You may need to create the libraries subfolder if its your first library. Restart the IDE.

Original BMP085 Repository (Deprecated)

This repository is not recommended for new projects, but is provided for reference sake for projects that made use of the BMP085 before the updated Unified driver was published

Github repository with library and code examples
(http://adafru.it/aKI)
To download click the DOWNLOADS button in the top right corner, rename the uncompressed folder Adafruit_BMP085. Check that the Adafruit_BMP085 folder contains Adafruit_BMP085.cpp and Adafruit_BMP085.h

Place the Adafruit_BMP085 library folder your /libraries/ folder. You may need to create the libraries subfolder if its your first library. Restart the IDE.

Buy a BMP085 Breakout

Buy a BMP085 Breakout (http://adafru.it/391)