BMP180 Arduino Tutorial with Library

Posted abril 14, 2012 by Love Electronics.

Introduction.

Measuring pressure is critical in many environments. Applications that immediately present themselves are pressure controlled environments like aircraft, or manufacturing processes like cooking or fabricating multilayer circuit boards; Did you know that pressure can be equally useful in calculating altitude (height)?

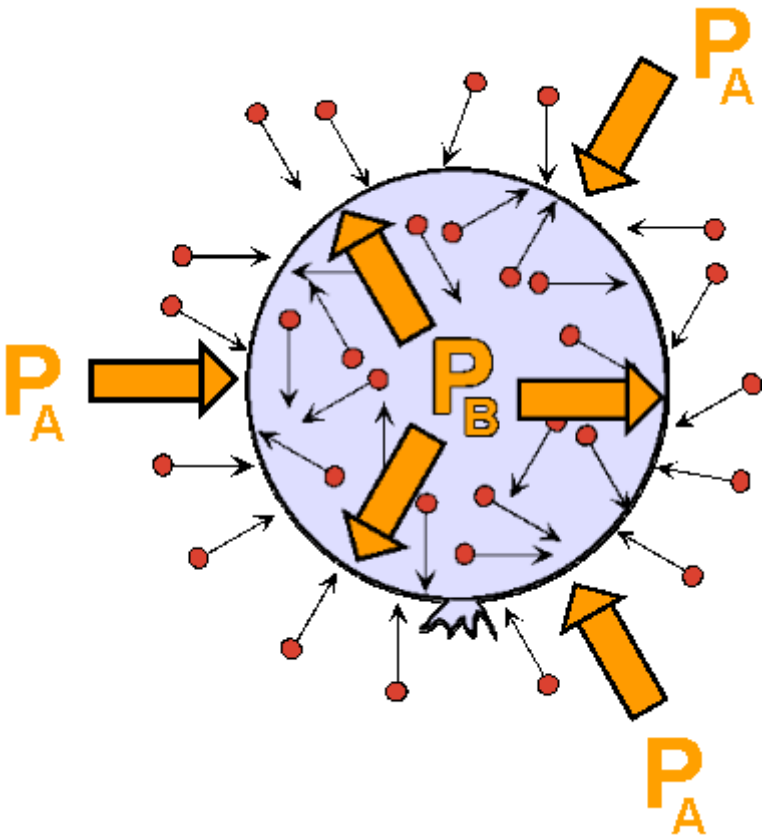
The purpose of this tutorial is to explain the idea behind pressure sensors; How they function and what formulas and code is required to use them. The sensor we will be using is the BMP180 Barometric Pressure Sensor from Bosch. We have this sensor mounted on a breakout board, with optional pull-up to make our job easier later on. At £20.99 it is a really cheap way to add

pressure sensing to your toolbox.

We will first go over the basic information about pressure and how it functions and is measured, and we will then build a custom altimeter using our BMP180 Breakout Board and an Arduino to communicate with the sensor.

What is pressure?

In order to use a pressure sensor, we need to understand the mechanics (at a simple level) of pressure. Pressure is simply the term for the amount of force applied perpendicular (at right angles) to the surface on a object. Simply put, it is the force that keeps a balloon expanded, and also the force that keeps it from exploding. There is air inside the balloon keeping it from collapsing, but there is also air outside the balloon, stopping it from expanding too far. Hopefully this graphic will indicate this:



In order to create a higher pressure environment, you just add more molecules to the same space. For instance, you can create liquid oxygen by squeezing more and more oxygen molecules into the same container and pressure increases until the oxygen gas condenses into liquid oxygen.

For a thorough example about how pressure works, see this page about pressure.

The BMP180 is a barometric pressure sensor, meaning it is a device used for measuring atmospheric pressure. In other words, it measures the amount of pressure you are currently feeling sitting at your computer.

Pressure is measured in many different units. It is measured in a force per area, such as lb per sq inch (psi). It can also be given in Pascals. This is more common when talking about atmospheric pressure. For example, the normal atmospheric pressure at sea level is 101.325 kPa.

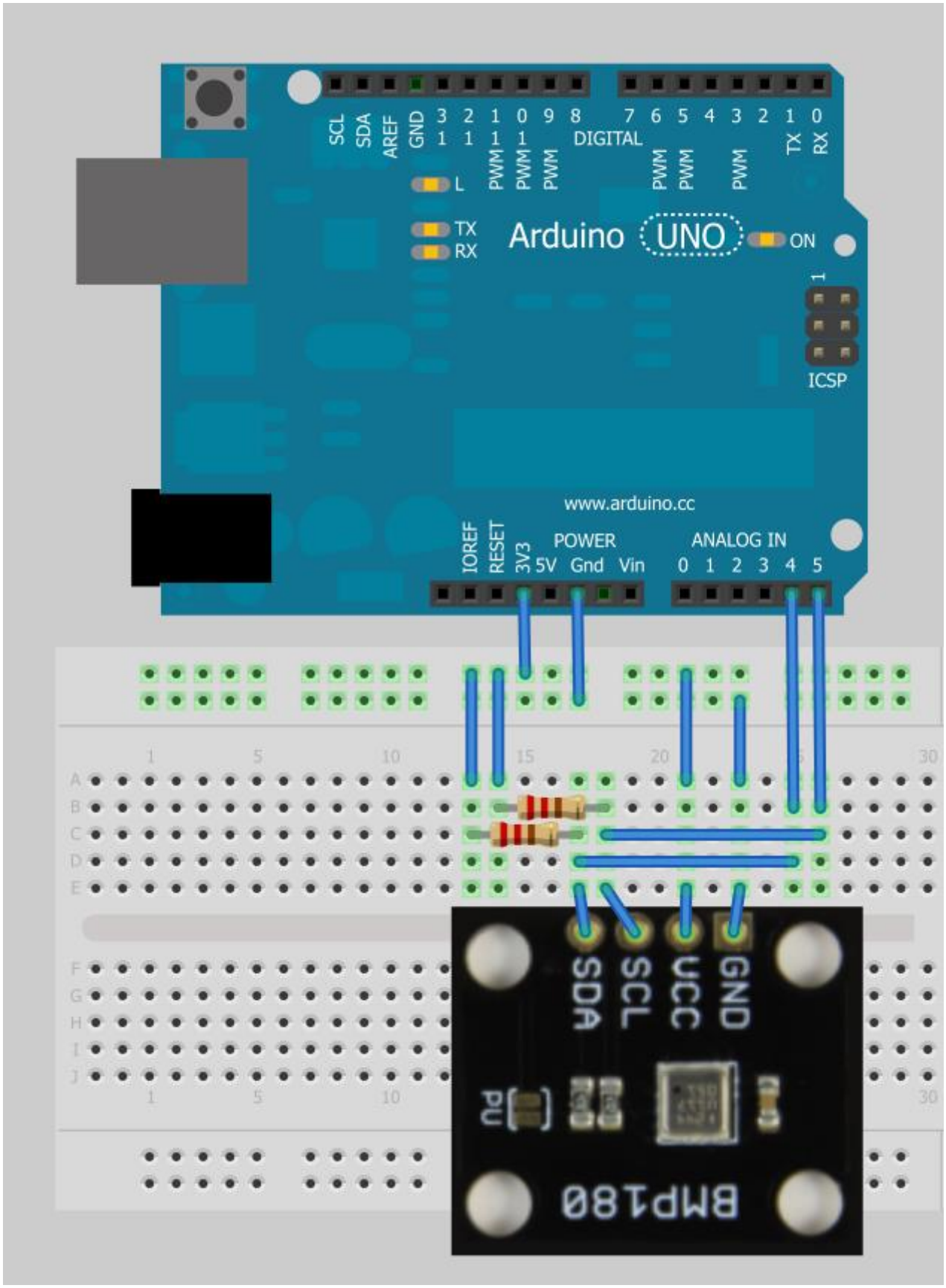
What is it good for?

As you probably know, the atmosphere is thinner the higher you go, until you end up with none, and your in space. Because of the fact that air is denser at ground level it also means the pressure at that height is greater than a location further up in the atmosphere. We see this in action later on in the tutorial, but an example is the air pressure at Mt. Everest. At sea level, atmospheric pressure is approximately 101.325 kPa, however at the summit of Mt. Everest the pressure drops to just 33.7 kPa! You can see from this example that by simply adding the scale of height to these measurements we can quickly determine the altitude of the sensed pressure.

Let's measure the pressure.

Now we know how pressure works we will need to construct a simple circuit to host our BMP180 Pressure Sensor in and connect it to our Arduino. The BMP180 communicates over the I2C bus to the Arduino, this requires use of the A4 and A5 (SDA, SCL) pins. Don't worry about current consumption. This nifty chip uses only 65mA whilst measuring, and just 0.1uA when idle!

Here is the circuit diagram, showing the connections between the sensor and the Arduino. A bonus point is that if you have one of our breakout boards you are able to skip the 4.7K pull up resistors on the SCL and SDA lines by just soldering the PU (pull up) jumper on the breakout board whilst your soldering on your headers.



We are also going to use the onboard LED to indicate we have succesfully connected the BMP180 pressure sensor to the Arduino. We use the onboard LED as it is easier than wiring up an LED and resistor just for this!

Once you're sure you have everything set up, we can start the next stage.

The BMP180 Library

We've created a useful Arduino library for you guys to use for this sensor. All you need to do to install it is download the zip package and extract it to the Arduino/Libraries folder on your computer. This library has been created for the new Arduino 1.0 environment, so if you are using an older version of Arduino you'll need to upgrade.
Download the Love Electronics BMP180 Arduino Library (for Arduino 1.0+) .

Once we have the library installed, we can begin programming! First of all we start a new sketch and add the following code. This code sets up our basic program, and creates an instance of the BMP180 pressure sensor. We will use this object to interact with the sensor later on in the program. We also check that we can connect to the sensor, and if so, light our LED.

```
// Include the Wire library for I2C access.
#include <Wire.h>
// Include the Love Electronics BMP180 library.
#include <BMP180.h>

// Store an instance of the BMP180 sensor.
BMP180 barometer;
// We are going to use the on board LED for an indicator.
int indicatorLed = 13;

void setup()
{
  // We start the serial library to output our messages.
  Serial.begin(9600);
  // We start the I2C on the Arduino for communication with the BMP180
sensor.
  Wire.begin();
  // Set up the Indicator LED.
  pinMode(indicatorLed, OUTPUT);
  // We create an instance of our BMP180 sensor.
  barometer = BMP180();
  // We check to see if we can connect to the sensor.
  if(barometer.EnsureConnected())
  {
    Serial.println("Connected to BMP180."); // Output we are connected to the
computer.
    digitalWrite(indicatorLed, HIGH); // Set our LED.
  }
  else
  {
    Serial.println("Could not connect to BMP180.");
    digitalWrite(indicatorLed, LOW); // Set our LED.
  }
}

void loop()
{
}
```


So once you have this uploaded to your Arduino your onboard LED should now be on (or you can check the Serial window) showing that you have correctly wired up the BMP180 sensor to

your Arduino. If your LED is not being lit, check the circuit diagram again, making sure you have remembered to add the pull up resistors, or soldered the PU jumper on the breakout board.

Once we have connected to the pressure sensor, the next stage is to initialize/configure it. Those great guys at Bosch have already calibrated the sensor before we even put it on a breakout board for you, meaning they've tuned added information we can retrieve whilst working with the measurements to make them accurate.

To complete this, we add the following lines to our setup() routine. We are going to insert just below the line Serial.println("Connected to BMP180.");

```
// When we have connected, we reset the device to ensure a clean start.
barometer.SoftReset();
// Now we initialize the sensor and pull the calibration data.
barometer.Initialize();
```

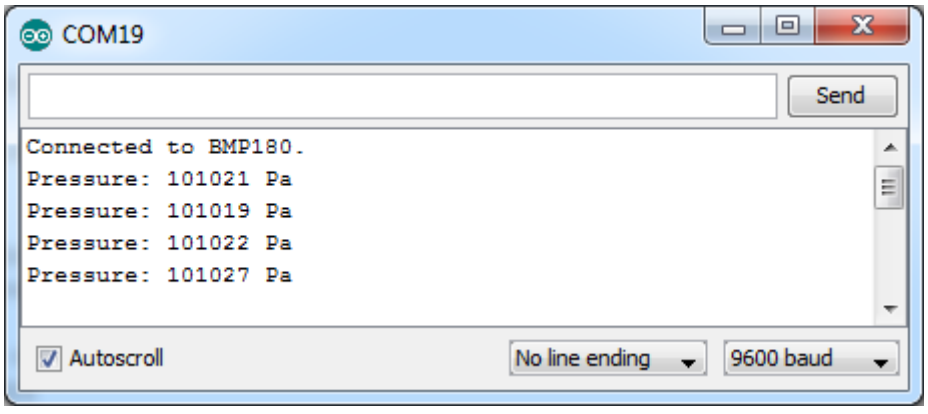

Great, now the BMP180 is set up and ready to take measurements. Now we can add some code into our loop() function to query the current pressure. The following loop() function asks the BMP180 for the pressure using the function GetPressure(). We then print this information to the Serial window.

```
void loop()
{
  if(barometer.IsConnected)
  {
    // Retrive the current pressure in Pascals.
    long currentPressure = barometer.GetPressure();

    // Print out the Pressure.
    Serial.print("Pressure: ");
    Serial.print(currentPressure);
    Serial.print(" Pa");

    Serial.println(); // Start a new line.
    delay(1000); // Show new results every second.
  }
}
```


You should be seeing something like so, your reported pressure will be different from mine depending on your location:



Let's recap at this point. We've understood what the principal is around pressure, how to use a sensor such as the BMP180 to measure this pressure. You've also followed the tutorial to

create a circuit to connect the BMP180 breakout board to an Arduino and created a simple sketch to measure the pressure.

Now comes the interesting part, calculating the altitude of our sensor. In order to calculate the altitude, we must first know what the current sea level pressure is at our location. You are looking for the Mean Sea Level (MSL) pressure. We need to feed this information into the Love Electronics BMP180 Arduino Library in order for it to calculate our altitude correctly. The best way I have found to do this is to get the information from your nearest airport in the UK, or if you are in the US, check weather.gov. If you cannot find your sea level pressure at the moment, use the average pressure of 101325 Pa. This will only affect your absolute air pressure, you will still be able to gauge changes in altitude with no change in accuracy.

Once you have found your sea level pressure, add it to the program before your setup() function, like so:

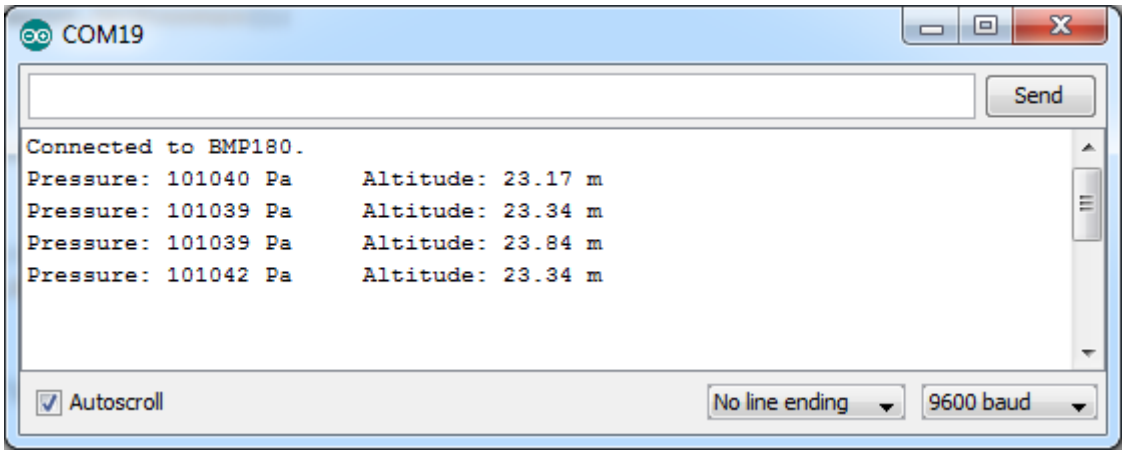
```
// Store the current sea level pressure at your location in Pascals.
float seaLevelPressure = 101325;
```


You then can use the BMP180 Arduino library to get your altitude! Lets put the following line into our loop() function, put these lines after Serial.print(" Pa");

```
    // Retrive the current altitude (in meters). Current Sea Level Pressure
is required for this.
    float altitude = barometer.GetAltitude(seaLevelPressure);

    // Print out the Altitude.
    Serial.print("\tAltitude: ");
    Serial.print(altitude);
    Serial.print(" m");
```


Congratulations! You've successfully attached the BMP180 to your Arduino, imported the Love Electronics BMP180 Arduino Library and then used the sensor to calculate your altitude! You could now log these readings to an SD card or similar and go on a road trip, and see how your altitude changes. Why not post your altitude to the comments and see how high we all are!? Here is what you should be seeing in your Serial window:



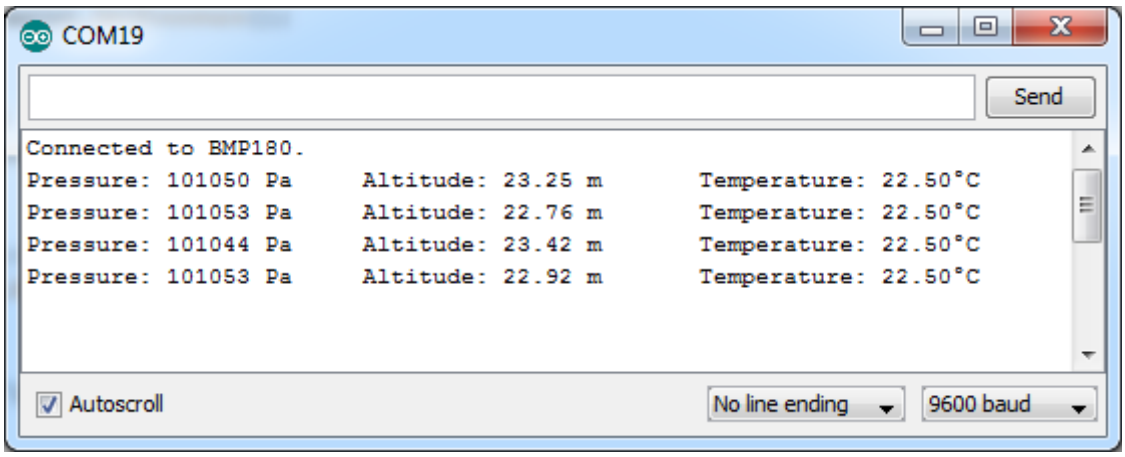
What you may have thought from the top description about pressure, is that it changes with temperature. This means in order to calculate our pressure correctly we need to know the temperature at which the measurement was taken. Luckily the BMP180 barometric pressure sensor has a temperature sensor inside. Inside our Love Electronics Arduino Library for the BMP180 Breakout Board we retrive the temperature when you ask for the pressure, so you

don't need to ask for it explicitly. However it's always nice to have more info. To add the temperature to our list of measurements, add this to the loop() function after Serial.print(" m");

```
    // Retrieve the current temperature in degrees Celsius.
    float currentTemperature = barometer.GetTemperature();

    // Print out the Temperature
    Serial.print("\tTemperature: ");
    Serial.print(currentTemperature);
    Serial.write(176);
    Serial.print("C");
```

view raw gistfile1.ino This Gist brought to you by GitHub.



Great! Now we are collecting and outputting all the information the BMP180 sensor can collect, using only a few lines of code thanks to the Love Electronics BMP180 Arduino Library.

I hope you've enjoyed this tutorial, if you'd like to try this out for yourself, why not order yourself a BMP180 Breakout Board for yourself, its just £20.99!

Please Like or Tweet this tutorial if you've found it useful and give us a helping hand!

Downloads:

Love Electronics BMP180 Arduino