MPL3115A2 Pressure Sensor Hookup Guide a learn.sparkfun.com tutorial

Available online at: http://sfe.io/t146

Contents

- MPL3115A2 Overview
- Hooking It Up
- Arduino Code
- Pressure vs Altimeter Setting
- Resources and Going Further

MPL3115A2 Overview

The MPL3115A2 is a low-cost, low power, highly accurate barometric pressure sensor. Use this sensor to detect changes in barometric pressure (weather changes) or for altitude (UAV controllers and the like). The sensor is *very* sensitive and capable of detecting a change of only 0.05kPa which equates to a 0.3m change in altitude.

SparkFun Altitude/Pressure Sensor Breakout - MPL3115A2

SEN-11084
$16.50
15
Favorited Favorite 57
Wish List

Things you should know about this sensor:

- Uses the I²C interface
- Only one sensor can reside on the I²C bus
- Uses the I²C repeated start condition. Arduino supports this, check if you're using a different microcontroller.
- Typical pressure accuracy of ±0.05kPa
- Typical altitude accuracy of ±0.3m
- Typical temperature accuracy of ±3C
- 3.3V sensor - use inline logic level converters or 330 ohm resistors to limit 5V signals
- Here's the datasheet
- Here's the schematic for the breakout board

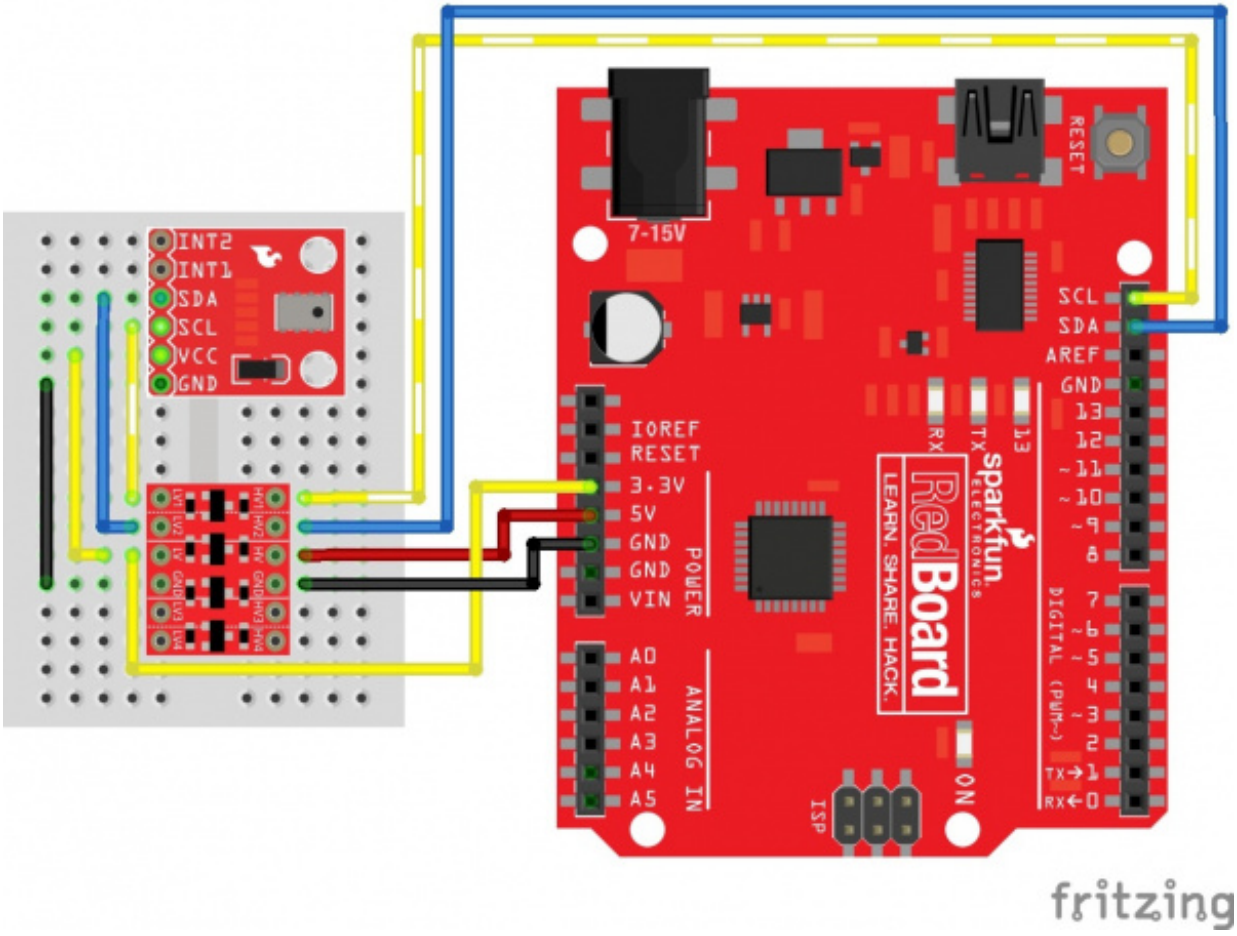
This sensor is ideal for environmental sensing, a weather station, or datalogging. It is a worthy replacement for the BMP085 and is more sensitive than theMPL115A1.

Suggested Reading

- Read about I²C!
- Using 3.3V sensors with 5V systems
- Installing an Arduino library
- What are pull-up resistors?
- Local Pressure vs Weather Station Pressure

Hooking It Up

Wiring up the MPL3115A2 pressure sensor is very easy! Aftersoldering the headers of your choice on the board, you'll need to convert the logic between the 5V and the sensor using a logic level converter.



You'll need 5v and 3.3V for VCC, one for GND, and two data lines for I²C communication from your Arduino. You may also use the A4 and A5 pins on older Arduino Boards that do not have SDA and SCL broken out.

Note: This breakout board has built-in 1kΩ pull up resistors for I²C communications. If you're hooking up multiple I²C devices on the same bus, you may need to disable the other resistors on the bus.
Note: If you have a RedBoard Qwiic, there is an alternative to use the built-in logic level converter using the qwiic adapter and cable or jumpers that can be used to adjust the voltage level to 3.3V.

SparkFun RedBoard Qwiic

DEV-15123
$21.50

19
Favorited Favorite 58
Wish List

SparkFun Qwiic Adapter

DEV-14495
$1.60
4
Favorited Favorite 64
Wish List

Qwiic Cable - Grove Adapter (100mm)

PRT-15109
$1.60
Favorited Favorite 18
Wish List

Hookup Table

With respect to the logic level converter, the pin connections starting from LV1 are listed in the table below.

MPL3115A2	Logic Level Converter *(Low Side)*	Logic Level Converter *(High Side)*	5V Arduino w/ Atmega328P
SCL	LV1	HV1	SCL *(or A5)*
SDA	LV2	HV2	SDA*(or A4)*
VCC	LV		3.3V

		HV	5V
GND	GND	GND	GND

Note: Not all microcontrollers use the same pin for SDA and SCL. If you are using a different architecture that is not the Atmega328P, make sure to edit the code accordingly if you use those pins instead.

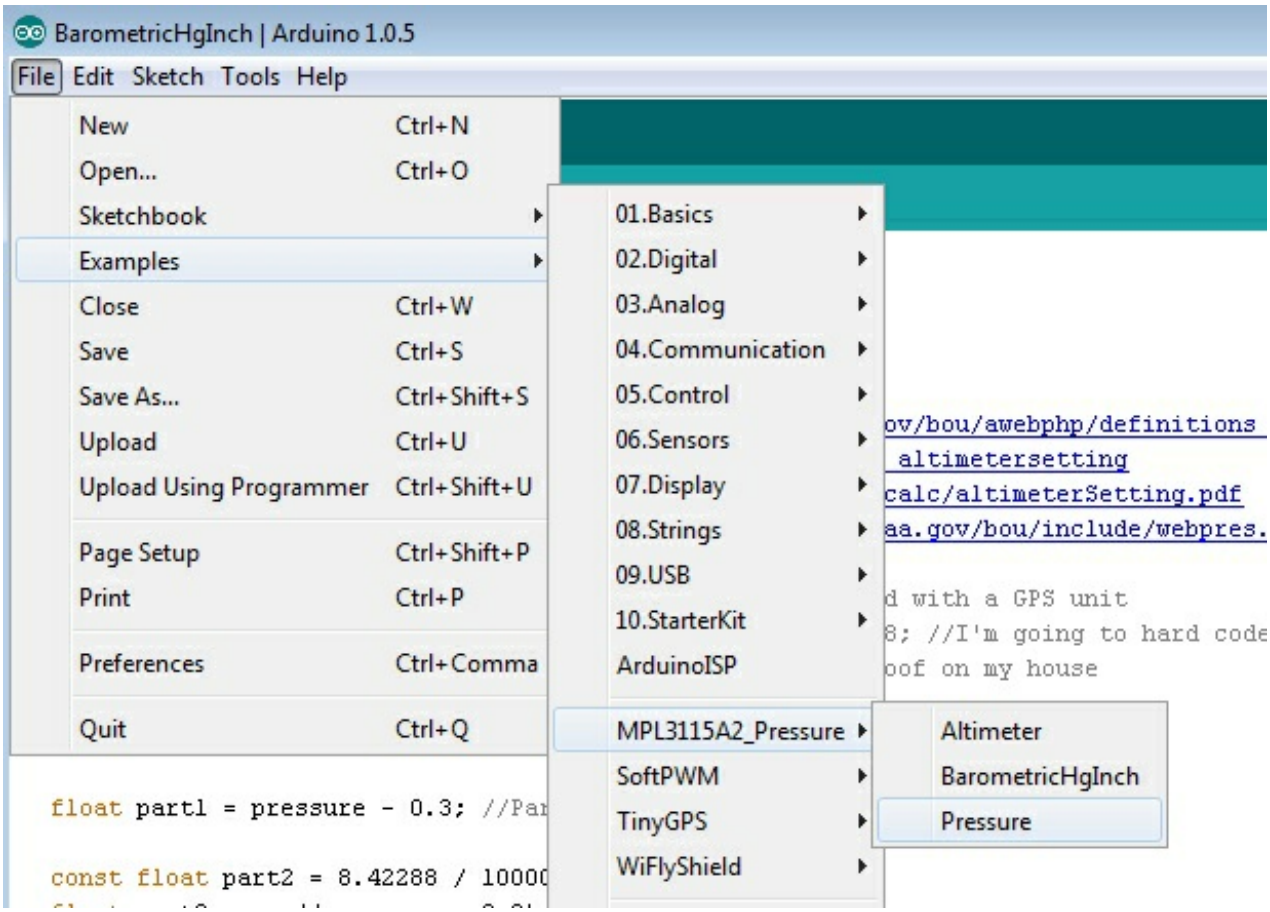
Arduino Code

The following Arduino example will get your sensor up and running quickly, and will show you current pressure in Pascals.

Note: This example assumes you are using the latest version of the Arduino IDE on your desktop. If this is your first time using Arduino, please review our tutorial on installing the Arduino IDE.

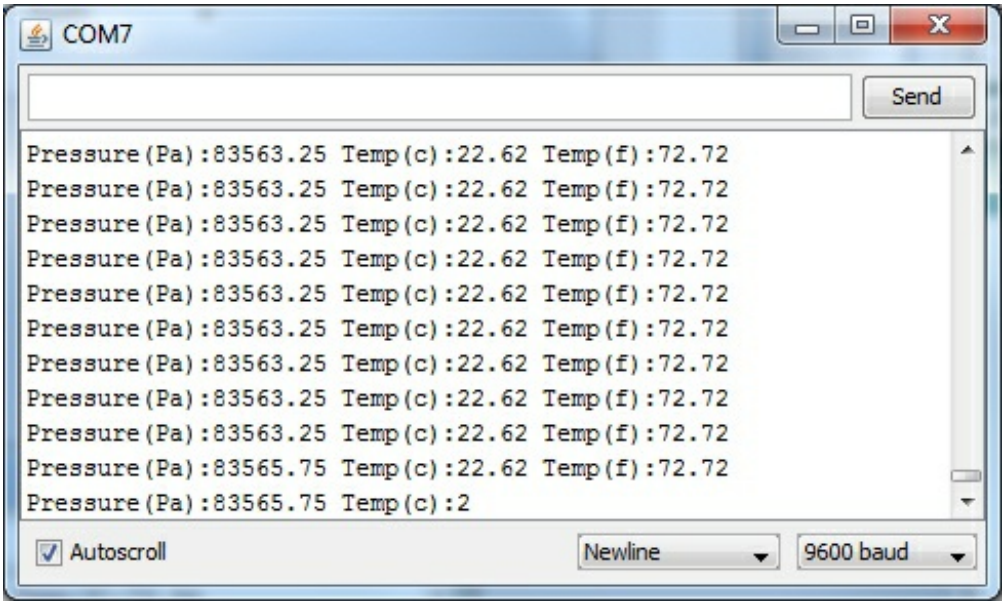
If you have not previously installed an Arduino library, please check out ourinstallation guide.

Download library here



Load the Pressure example

Once the library is installed, open Arduino, and expand the examples menu. You should see the MPL3115A2_Pressure sub-menu. Load the "Pressure" example onto the Arduino. Open the serial terminal at 9600bps. You will see the current barometric pressure and temperature in the room!



Pressure readings!

Load the BarometricHgInch example for an example that coverts pressure from Pascals to inches of mercury, altimeter setting adjusted. This type of pressure reading is used in the USA on Wunderground for home weather stations and aircraft.

Load the Altimeter example for an example that coverts pressure to current altitude in feet (or meters).

Explanation of Functions

The library and example code demonstrate the most popular functions supported by the MPL3115A2. Here is an explanation of all the available functions in the library:

- **myPressure.begin()** gets sensor on the I²C bus.
- **myPressure.readAltitude()** returns a float with meters above sea level. Ex: 1638.94
- **myPressure.readAltitudeFt()** returns a float with feet above sea level. Ex: 5376.68
- **myPressure.readPressure()** returns a float with barometric pressure in Pa. Ex: 83351.25
- **myPressure.readTemp()** returns a float with current temperature in Celsius. Ex: 23.37
- **myPressure.readTempF()** returns a float with current temperature in Fahrenheit. Ex: 73.96
- **myPressure.setModeBarometer()** puts the sensor into Pascal measurement mode.
- **myPressure.setModeAltimeter()** puts the sensor into altimetry mode.
- **myPressure.setModeStandy()** puts the sensor into Standby mode. Required when changing CTRL1 register.
- **myPressure.setModeActive()** starts taking measurements!
- **myPressure.setOversampleRate(byte)** sets the # of samples from 1 to 128. See note below *
- **myPressure.enableEventFlags()** sets the fundamental event flags. Required during setup.

When you call the readAltitude, readAltitudeFt, readPressure, or readTemp you will get a float with the sensor reading or an error code:

- 1638.94 is an example of a valid reading.
- -999 indicates that I2C timed out (512ms max). Check your connections.

Table 59. System Output Sample Rate Selection

OS2	OS1	OS0	Oversample Ratio	Minimum Time Between Data Samples
0	0	0	1	6 ms
0	0	1	2	10 ms
0	1	0	4	18 ms
0	1	1	8	34 ms
1	0	0	16	66 ms
1	0	1	32	130 ms
1	1	0	64	258 ms
1	1	1	128	512 ms

Oversample settings

- **setOversampleRate(byte)** receives a value from 0 to 7. Check table 59 above. Allows the user to change sample rate from 1 to 128. Increasing the sample rate significantly decreases the noise of each reading but increases the amount of time to capture each reading. A oversample of 128 will decrease noise to 1.5Pa RMS but requires 512ms per reading. The datasheet recommends oversample of 128 for basic applications.

The MPL3115A2 has a large number of features. Checkout thedatasheet for more info. This library covers the fundamentals. Help us out! Please add or suggesting more features on the MPL3115A2 github repo.

Pressure vs Altimeter Setting

If you grabbed a few pressure readings and became confused when you checked your local weather conditions, you're not alone. The absolute pressure that the MPL3115A2 pressure sensor outputs is not the same as what weather stations refer to as pressure. Weather stations report pressure in lots of different units:

- millimeters Mercury (mmHg)
- inches Mercury (inHg)
- millibars or hectopascals (hPa)
- pounds per square inch
- atmospheres (Atm)
- kilogram per centimeter
- inches of water

In barometer mode, the MPL3115A2 outputs pressure readings in Pascals. This is most closely

related to millibars or hectopascals. But, why does the sensor not agree with the station around the corner? This is because many stations report pressure in a few different formats. Have a look at all these numbers for the Boulder/Denver area. The key is that your local weather station is probably reporting the *Altimeter setting*.

Thank you National Oceanic and Atmospheric Administration (NOAA)! Did you know they're headquartered here in Boulder, CO?

- **Station pressure** - This is the pressure that is observed at a specific elevation and is the true barometric pressure of a location.
- **Altimeter setting** - This is the pressure reading most commonly heard in radio and television broadcasts. It is not the true barometric pressure at a station. Instead it is the pressure "reduced" to mean sea level using the temperature profile of the "standard" atmosphere, which is representative of average conditions over the United States at 40 degrees north latitude.
- **Mean sea level pressure** - This is the pressure reading most commonly used by meteorologists to track weather systems at the surface. Like the altimeter setting, it is a "reduced" pressure, which uses observed conditions rather than "standard" conditions to remove the effects of elevation from pressure readings.

The calculation to get from Pascals to 'Altimeter setting' is a bit gnarly:

$$h_m = 0.3048 \times h_{ft}$$

$$Alt = (P_{mb} - 0.3) \times \left(1 + \left(\left(\frac{1013.25^{0.190284} \times 0.0065}{288}\right) \times \left(\frac{h_m}{(P_{mb} - 0.3)^{0.190284}}\right)\right)\right)^{\frac{1}{0.190284}}$$

Formula to convert Pascal pressure to Altimeter setting

Grab the full formula here and give this great Altimeter setting calculator a try. This formula relies on two things: knowing the current pressure in milibars and knowing the height above sea level that the pressure was read. We recommend you capture altitude using a local survey point or a GPS receiver.

If you installed the MPL3115A2 library, you should also have the *BarometricHgInch* example sketch under the Examples->MPL3115A2_Pressure menu under the Arduino IDE. We didn't build this calculation into the library because it could potentially chew up a lot of RAM and code space calculating all the floating point math. But, if you're doing home weather station calculations, this should get you started.

Resources and Going Further

Now that you've got barometric pressure sensing under your belt, consider checking out the following projects and products:

- MPL3115A2 Breakout Schematic (PDF)

- MPL3115A2 Datasheet
- GitHub
  - Product Repo
  - Arduino Library
- SparkFun Simple Sketch Demo

Looking for ideas to use the MPL3115A2 pressure sensor? Well you could:

- Combining with an Electric Imp to create a wireless pressure sensor
- Combining multiple pressure sensors to create a tactile robotic hand. Also see TakkTile.
- Checkout the Weather Shield for the MPL3115A2 combined with many other sensors

---