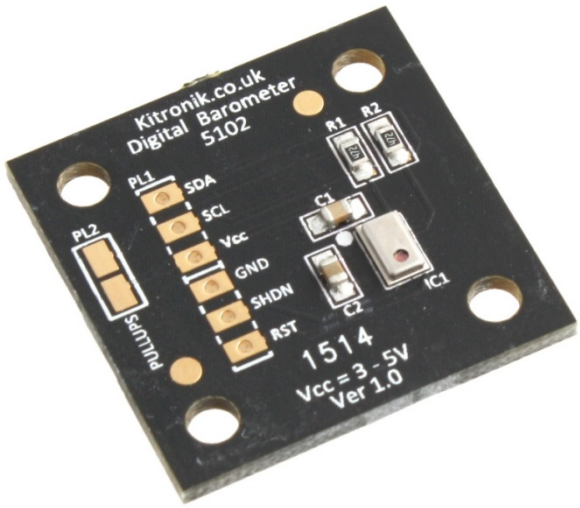
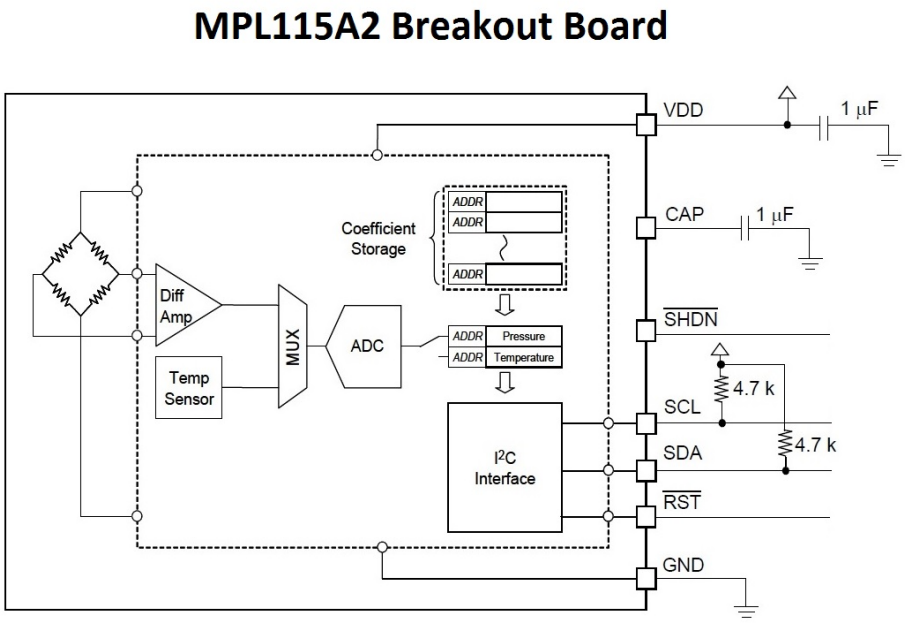
Barometric Sensor Breakout - 5102

Circuit Overview

This board uses a MPL115A2 digital barometer which can provide a temperature compensated pressure reading. The sensor can measure pressures between 50 kPa to 115 kPa Absolute Pressure with an accuracy of ±1 kPa. The board supports a wide range of input voltages (2.4V – 5.5V) and uses I2C. When selecting an input voltage to use, keep in mind that pulling the SHDN and RST pins above the input voltage will cause malfunctions. On board IIC bus pull-ups are included and can be enabled by attaching a jumper across PL2.



Block Diagram



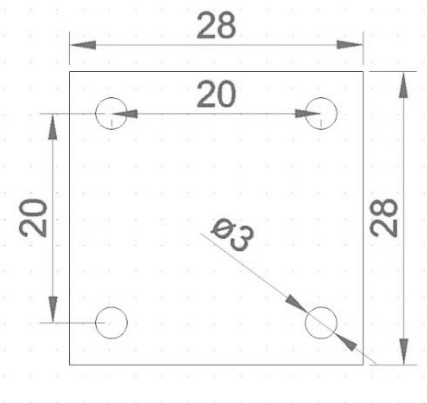
Electrical Characteristics

	Min	Typical	Max
Supply Voltage	2.4V	3.3V	5.5V
Current Consumption (sampling)		5uA	6uA
Current Consumption (SHDN)			1uA
Range	50kPa		115kPa
Resolution		0.15kPa	
Accuracy			±1 kPa
Conversion time		1.6ms	3ms
Wakeup time		3ms	5ms

Pinout

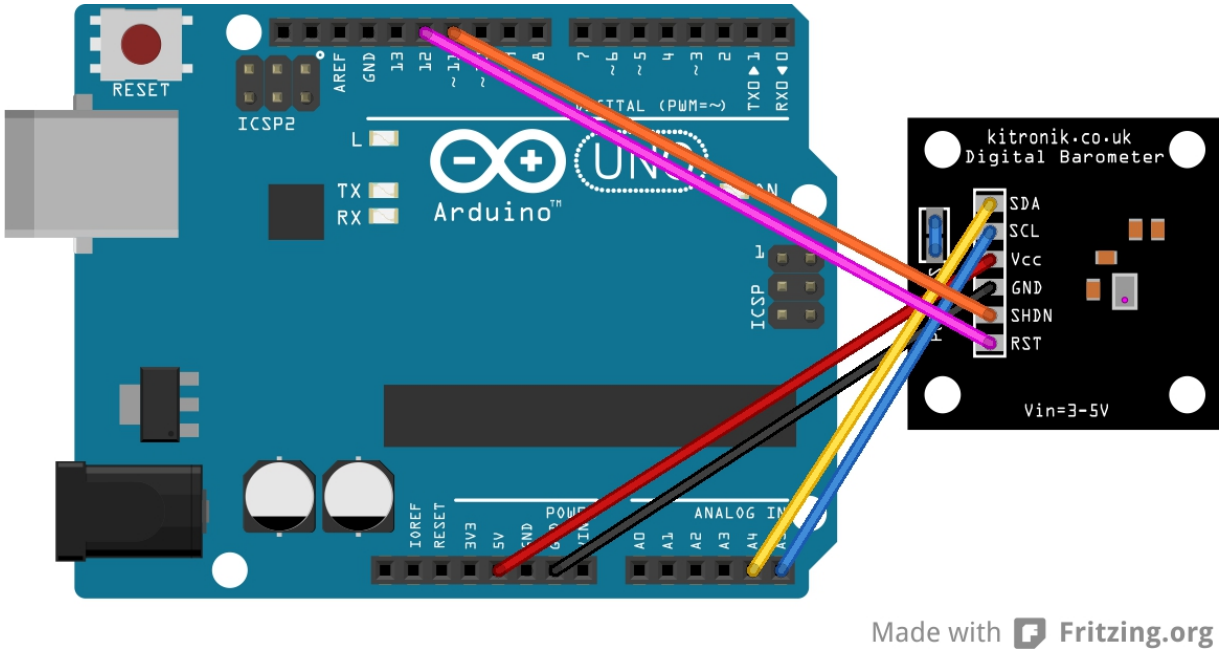
SDA	I2C data	
SCL	I2C clock	
Vcc	Supply voltage	2.4-5.5V
GND	Ground pin	0V
SHDN	Shutdown pin	Low = shutdown High = active (must not exceed Vcc)
RST	Reset Pin	Low = reset High = normal (must not exceed Vcc)
PL2	Pull-up jumper	Connect pads with solder bridge to enable pull-ups

Mechanical Information



All dimensions in mm. Tolerance 0.1mm. All mounting holes are on 10mm spacings.

Arduino Uno Connections



Barometric Sensor	Arduino Uno
SDA	A4*
SCL	A5*
Vcc	5V or 3.3V (to match IOREF)
GND	GND
SHDN	Pin 11
RST	Pin 10

*SDA and SCL pins vary between Arduino models. If you wish to check which pins your Arduino uses for SDA/SCL you can visit http://arduino.cc/en/reference/wire

Arduino Sketch

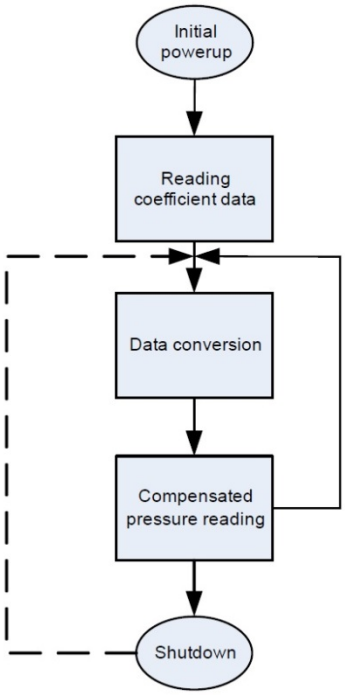
The sketch for this board is quite long so it is provided in a .zip.

In simple terms the program retrieves stored calibration information, and then uses it along with a pair of readings taken by the device to calculate pressure. The time taken for a reading to be taken and calculated is around 7.5ms at 5V

If ran "as-is" the program will simply output the pressure every 5 seconds in kPa to the serial monitor. The refresh rate can be altered by changing the value of the variable 'refreshtime', which is measured in milliseconds and is set to 50,000 by default.

If you are interested in how this is calculated or are looking to debug a problem you might have, you can increase the amount of data displayed on the serial monitor. To do this simply look at the 2nd line of code "//#define DEBUG" and delete the "//". Putting them back in again will hide the information.

Function Overview



The MPL115A2 works in several steps:

The initial power-up causes all circuit elements to become active. I2C port pins are high impedance and associated registers are cleared. The device then enters standby mode.

The user then accesses the ROM in the device and reads the coefficients, these are then usually stored in the microcontroller, but can be re-read at any time.

The user sends a CONVERT command to the device to begin data conversion, when this is complete the raw data is stored in the device's pressure and temperature ADC registers.

The microcontroller then uses the coefficients and raw pressure and temperature data to give a compensated pressure reading by using a compensation formula shown below.

$$Pcomp = a0 + (b1 + c12 \cdot Tadc) \cdot Padc + b2 \cdot Tadc$$

Padc is the 10-bit pressure ADC output of the MPL115A
Tadc is the 10-bit temperature ADC output of the MPL115A
a0 is the pressure offset coefficient
b1 is the pressure sensitivity coefficient
b2 is the temperature coefficient of offset (TCO)
c12 is the temperature coefficient of sensitivity (TCS)

$$Pressure(kPa) = Pcomp\left[\frac{115-50}{1023}\right]+50$$

I2C Commands (simplified for communication)

Device Address + write bit “To Write”	0xC0
Device Address + read bit “To Read”	0xC1
Command to Write “Convert Pressure and Temperature”	0x12*
Command to Read “Pressure ADC High byte”	0x00
Command to Read “Pressure ADC Low byte”	0x01
Command to Read “Temperature ADC High byte”	0x02
Command to Read “Temperature ADC Low byte”	0x03
Command to Read “Coefficient data byte a0_MSB”	0x04
Command to Read “Coefficient data byte a0_LSB”	0x05
Command to Read “Coefficient data byte b1_MSB”	0x06
Command to Read “Coefficient data byte b1_LSB”	0x07
Command to Read “Coefficient data byte b2_MSB”	0x08
Command to Read “Coefficient data byte b2_LSB”	0x09
Command to Read “Coefficient data byte c12_MSB”	0x0A
Command to Read “Coefficient data byte c12_LSB”	0x0B

***The command byte needs to be paired with a 0x00 as part of the I2C exchange to complete the passing of Start Conversions.**

Glossary of terms

kPa – kilopascal, standard unit for pressure equal to 1000 pascals

I2C/IIC – inter-integrated circuit, a computer bus invented by Philips used for attaching low-speed peripherals to a motherboard, embedded system or other electronic device

SDA – Serial Data

SCL – Serial Clock

SHDN – Shutdown

RST – Reset

MUX – A multiplexer (or mux) is a device that selects one of several analogue or digital input signals and forwards the selected input into a single line.

ADC – Analogue Digital Converter

Vcc – Supply Voltage

Troubleshooting

If the pressure is continuously reading around 47kPa it is possible that the reset and shutdown pins are being held above Vcc. If SHDN and RST are held at 5V then Vcc must be 5V.

If you are receiving no data at all then make sure you have pull-up resistors on the SCL and SDA lines. Pull-ups are included on the breakout board and can be included by making a solder link across the two pads of PL2.