Embedded Workshop Oct 25, 2017

Rusty Cain

While you are waiting for the Workshop to begin…

Make sure you are connected to the local Wifi Guest Password: **Welcome2DMS**

Arduino Uno & Wemos D1 Mini

1. **Download Arduino Version 1.8.1 or newer.**
   Add the Wire.h library
   Add the Adafruit_BMP085.h library Version 1.0.0
   Sketch - Include Libraries - Manage Libraries
2. **Install Wemos software in Arduino IDE**
   Wemos Main Page: https://www.wemos.cc/
   Installing Wemos in Arduino IDE:
   https://wiki.wemos.cc/tutorials:get_started:get_started_in_arduino
3. **Copy Programs off the USB Memory stick.**
   Project1,

Parts Needed:

Arduino Uno or Wemos D1 Mini
Breadboard & Wires,

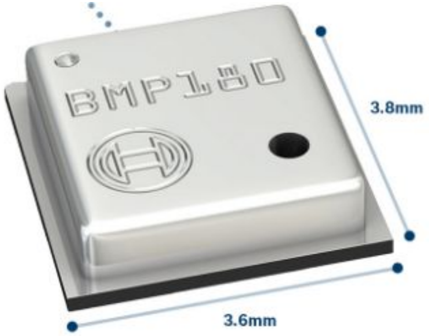
Raspberry Pi 3

1. **Download the following Files:**

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

BMP180 Overview

The BMP180 is the digital barometric pressure sensor from Bosch that measures barometric pressure, altitude and temperature. Pressure and temperature data are Provided as 16 bit values.



Datasheet:

https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BMP180-DS000-121.pdf

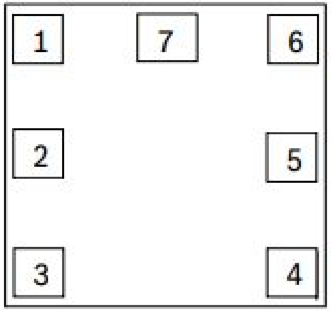
Driver:

https://github.com/BoschSensortec/BMP280_driver

Other sensors in the same category

BMP180 Overview

Pin 1 CSB N/C -SPI Chip Select
Pin 2 VDD
Pin 3 VDDIO Digital Power Sup.
Pin 4 N/C SDO -SPI
Pin 5 SCL I2C Clock
Pin 6 SDA I2C Data
Pin 7 GND



SDA & SCL need 4.7K pull up resistors.

Note: I2C address Hex 77 Dec 119

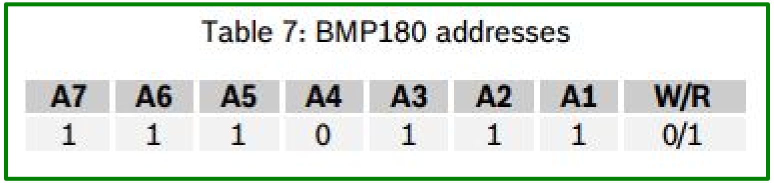
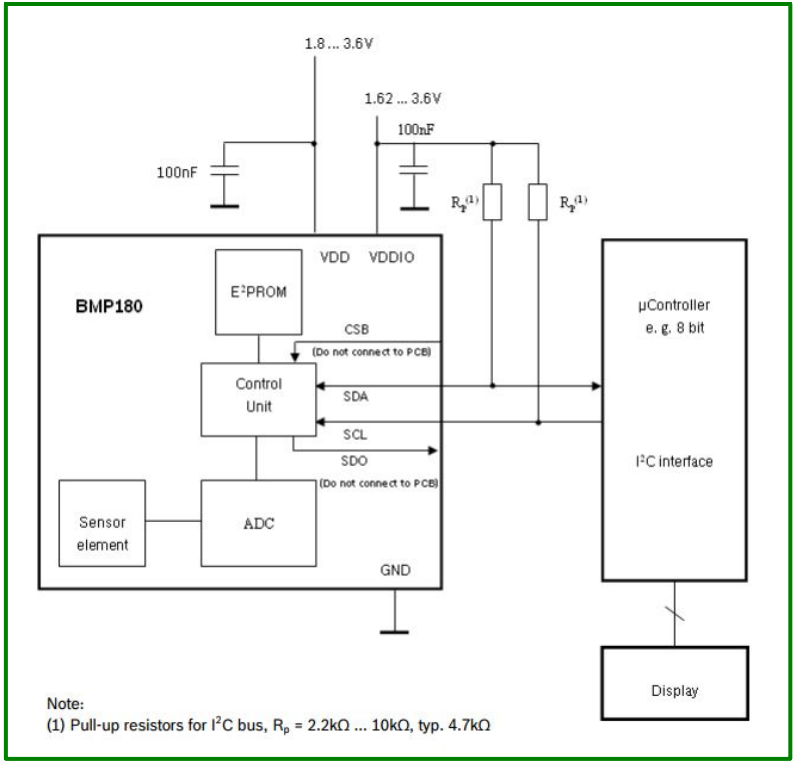
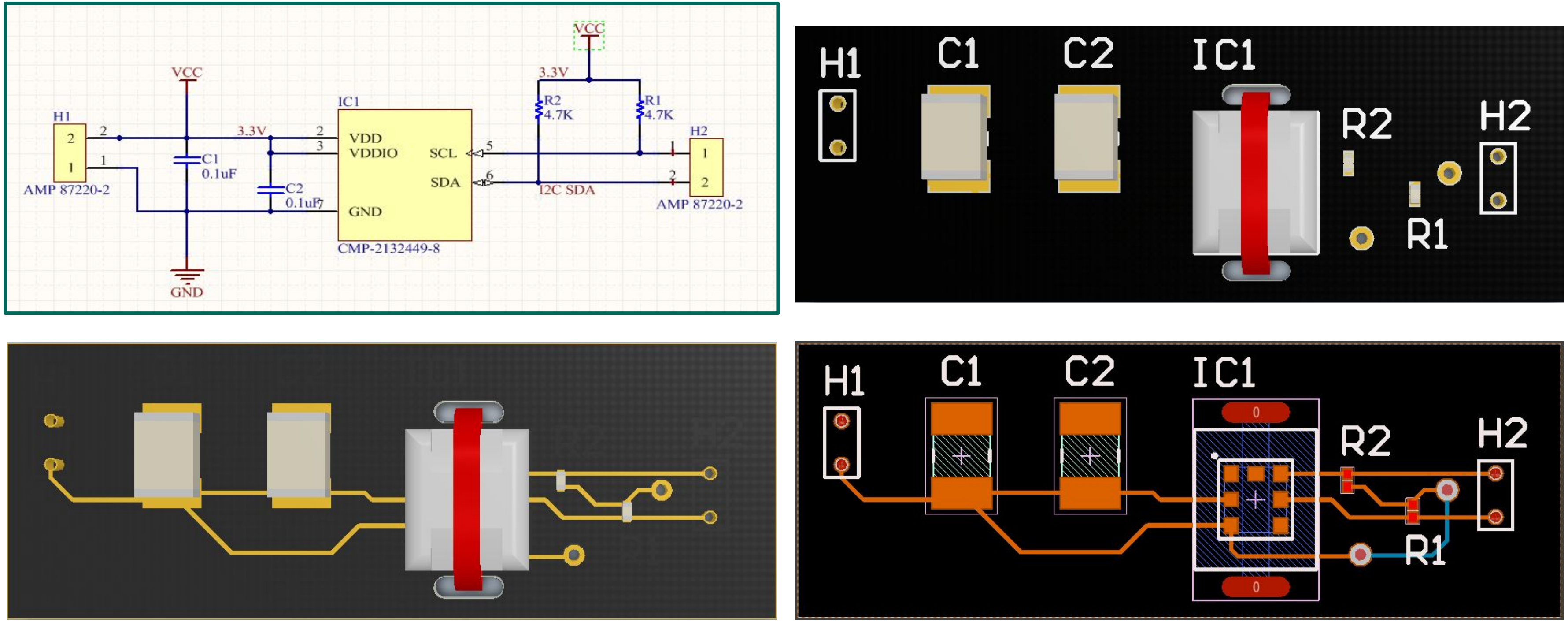


Table 7: BMP180 addresses

A7	A6	A5	A4	A3	A2	A1	W/R
1	1	1	0	1	1	1	0/1



Note:
(1) Pull-up resistors for I²C bus, R_p = 2.2kΩ ... 10kΩ, typ. 4.7kΩ



Design your own BMP180 Sensor PCB

Arduino Uno - BMP180 Connections

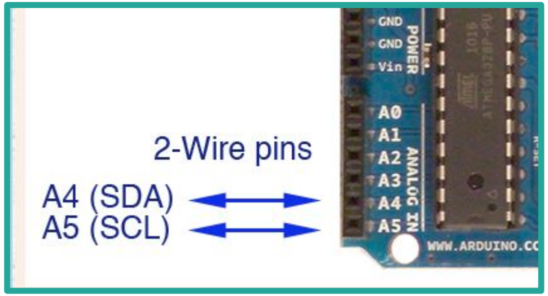
Connect the **VCC** pin to a **3.3V** power source.

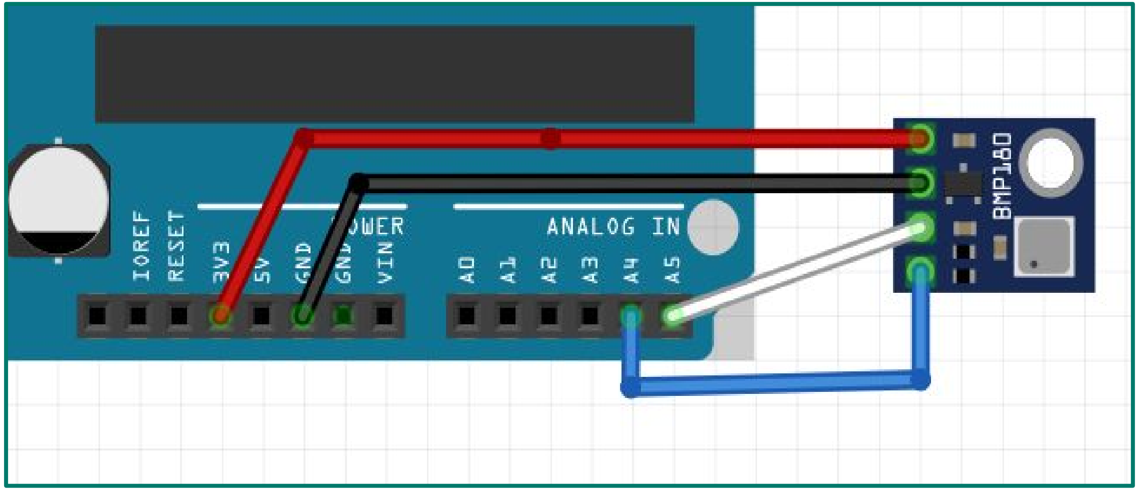
The BMP085 sensor breakout cannot be used with anything higher than 3.3V so don't use a 5V supply!

The BMP180 sensor board has a 3.3V regulator so you can connect it to either 3.3V or 5.0V.

Connect **GND** to the ground pin.

Analog port 4 (A4) = SDA (serial data)
Analog port 5 (A5) = SCL (serial clock)





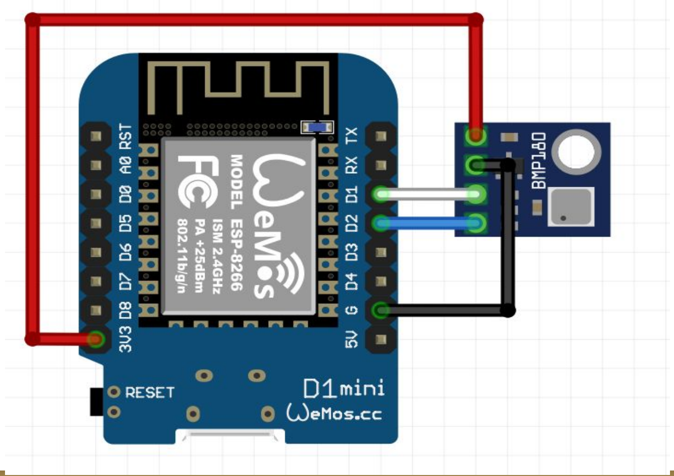
Wemos - BMP180

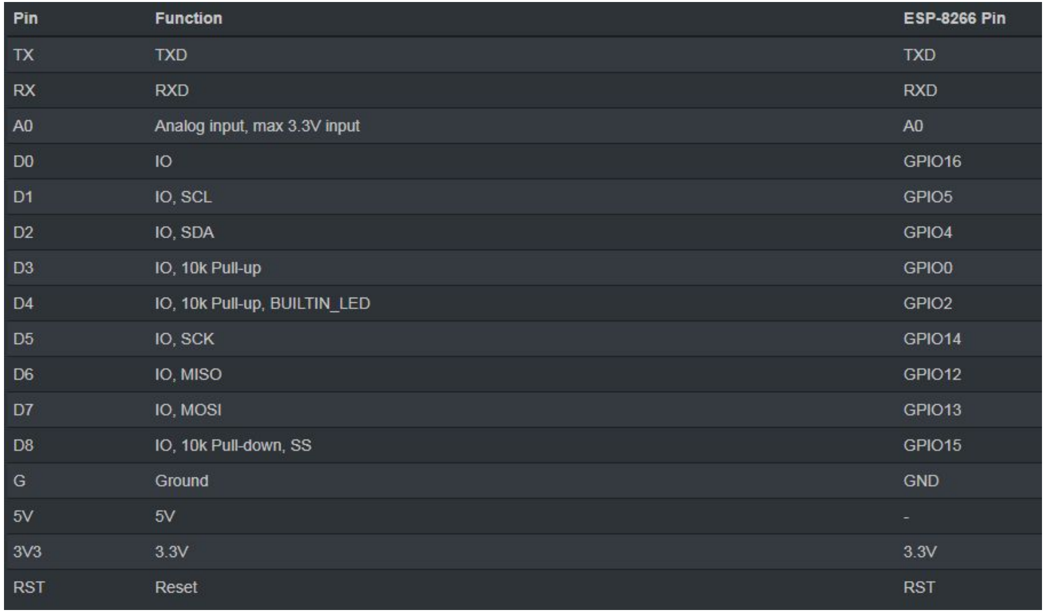
Installing Wemos in Arduino IDE:

https://wiki.wemos.cc/tutorials:get_started:get_started_in_arduino

In order to use the I²C interface we need to include the Arduino standard Wire library `#include <Wire.h>`

D1 Mini	**BMP180**
(D1) =	SCL (serial clock)
(D2) =	SDA (serial data)
3V3 =	VCC
G =	GND





Pin	Function	ESP-8266 Pin
TX	TXD	TXD
RX	RXD	RXD
A0	Analog input, max 3.3V input	A0
D0	IO	GPIO16
D1	IO, SCL	GPIO5
D2	IO, SDA	GPIO4
D3	IO, 10k Pull-up	GPIO0
D4	IO, 10k Pull-up, BUILTIN_LED	GPIO2
D5	IO, SCK	GPIO14
D6	IO, MISO	GPIO12
D7	IO, MOSI	GPIO13
D8	IO, 10k Pull-down, SS	GPIO15
G	Ground	GND
5V	5V	-
3V3	3.3V	3.3V
RST	Reset	RST

BMP180 using Adafruit Sensor Libraries

Adafruit states the following for calculating altitude:

> 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 degrees celsius.
> 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. Use your local weather website to find the current SLP value.

In order to use the I^{2}C interface we need to include the Arduino standard Wire library.

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

We also need to include the Adafruit Library.

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

```
#include <Wire.h> //  Include Arduino I2C library
#include <Adafruit_BMP085.h>
```

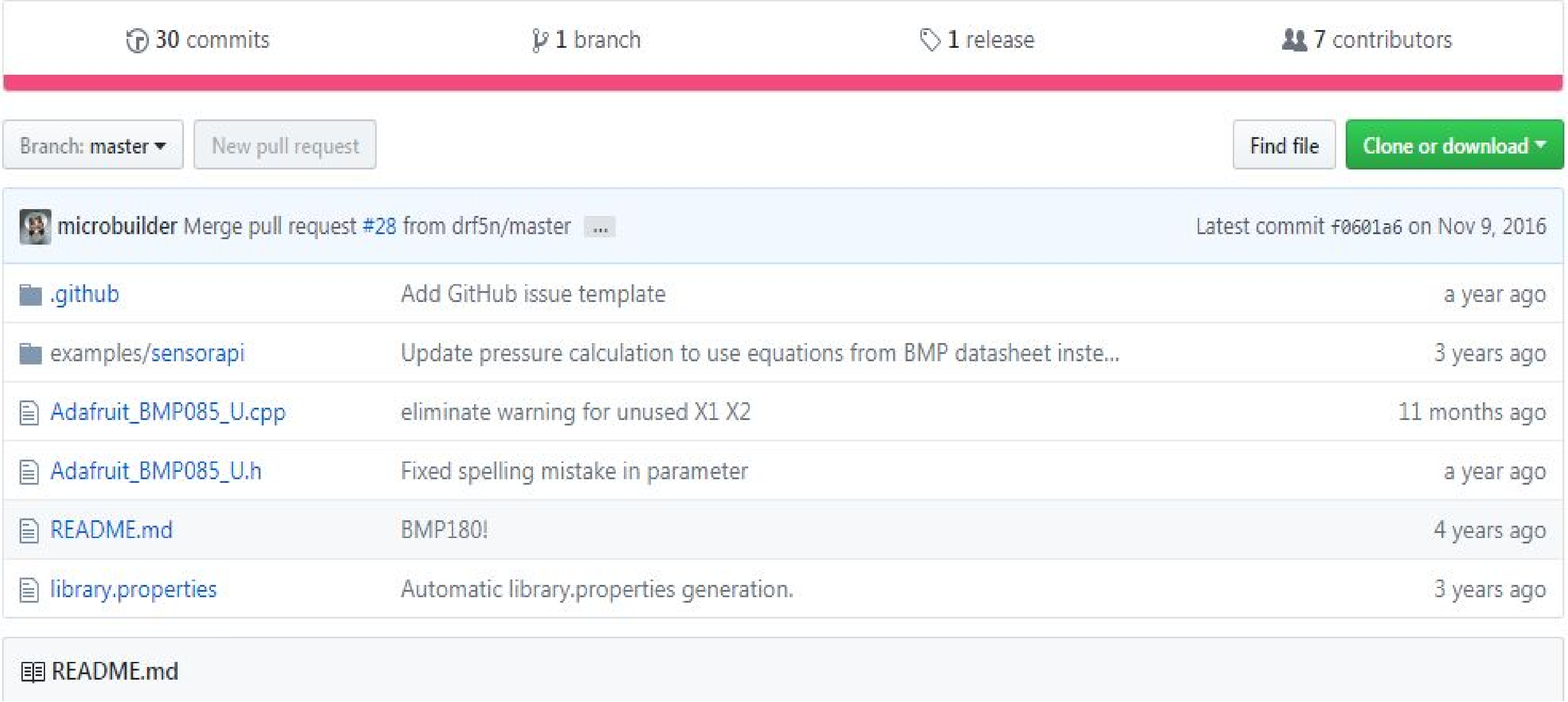
Adafruit BMP085 Library Version 1.0.0 or

Adafruit BMP085 Unified Version 1.0.0?

The same code is used for both the BMP085 and BMP180 (they are compatible!)

You can find the Arduino library repository on github

To install it, click this button to download the compressed ZIP file then install it.



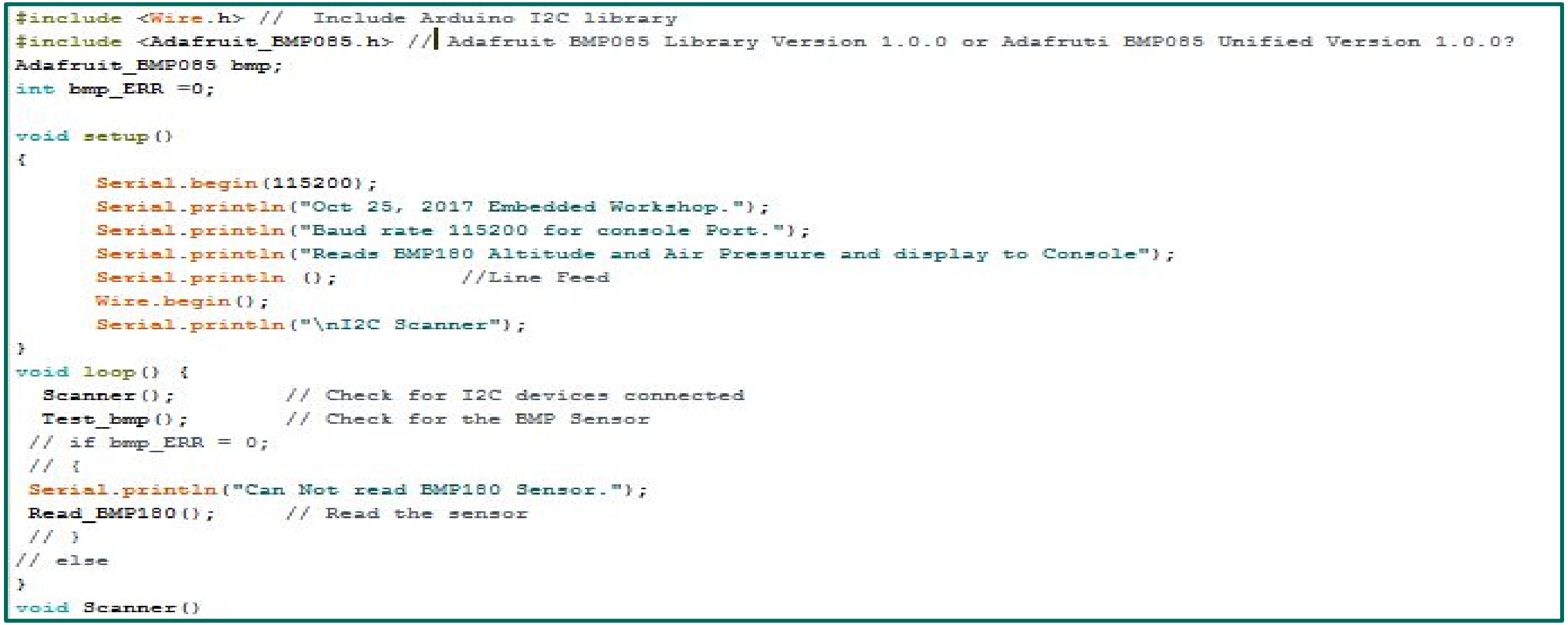
microbuilder Merge pull request #28 from drf5n/master … Latest commit f0601a6 on Nov 9, 2016

.github	Add GitHub issue template	a year ago
examples/sensorapi	Update pressure calculation to use equations from BMP datasheet inste…	3 years ago
Adafruit_BMP085_U.cpp	eliminate warning for unused X1 X2	11 months ago
Adafruit_BMP085_U.h	Fixed spelling mistake in parameter	a year ago
README.md	BMP180!	4 years ago
library.properties	Automatic library.properties generation.	3 years ago

README.md

Download Arduino Uno Workshop Code

Go to: www.the-cains-group.net
Select: **October 2017 Workshop - How to integrate a Bosch BMP180 I2C.**



```
#include <Wire.h> //  Include Arduino I2C library
#include <Adafruit_BMP085.h> //| Adafruit BMP085 Library Version 1.0.0 or Adafruti BMP085 Unified Version 1.0.0?
Adafruit_BMP085 bmp;
int bmp_ERR =0;

void setup()
{
      Serial.begin(115200);
      Serial.println("Oct 25, 2017 Embedded Workshop.");
      Serial.println("Baud rate 115200 for console Port.");
      Serial.println("Reads BMP180 Altitude and Air Pressure and display to Console");
      Serial.println ();          //Line Feed
      Wire.begin();
      Serial.println("\nI2C Scanner");
}
void loop() {
  Scanner();         // Check for I2C devices connected
  Test_bmp();        // Check for the BMP Sensor
 // if bmp_ERR = 0;
 // {
 Serial.println("Can Not read BMP180 Sensor.");
 Read_BMP180();      // Read the sensor
 // }
// else
}
void Scanner()
```

Sample output from Workshop Program

```
Oct 25, 2017 Embedded Workshop.
Baud rate 115200 for console Port.
Reads BMP180 Altitude and Air Pressure and display to Console

I2C Scanner
Scanning...
I2C device found at address 0x77  !
done

BMP Sensor Connected and Working

Temperature = 26.50 °C
Pressure = 100348 Pa
Altitude = 81.24 meters
Altitude = 265.44 Feet
Pressure at sea level (calculated) = 100363 Pa
Real altitude = 95.19 meters
```

Note: I2C address Hex 77 Dec 119

Wemos - BMP180

```
#include <Wire.h>
#include <Adafruit_BMP085.h>
Adafruit_BMP085 bmp;
void setup()
{
  Serial.begin(9600);
  //Wire.begin (1, 2);
  if (!bmp.begin())
  {
    Serial.println("Could not find BMP180 or BMP085 sensor at 0x77");
    while (1) { }
  }
}
```

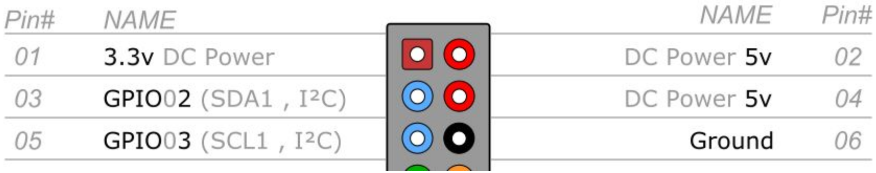
```
void loop()
{
  Serial.print("Temperature = ");
  Serial.print(bmp.readTemperature());
  Serial.println(" Celsius");
  Serial.print("Pressure = ");
  Serial.print(bmp.readPressure());
  Serial.println(" Pascal");
   Serial.println();
  delay(5000);
}
```

Code Resource:
http://www.esp8266learning.com/wemos-mini-bmp180-sensor-example.php

Raspberry Pi BMP180 Hardware Setup

The I2C bus allows multiple devices to be connected to your Raspberry Pi each with a unique address. Running the following commands in the Terminal to install the i2c-tools utility.

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

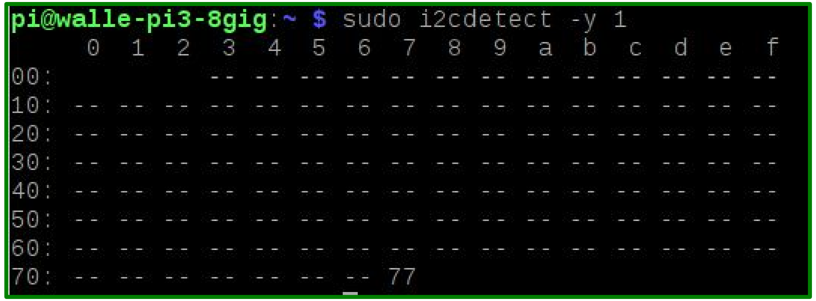


Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I²C)	DC Power 5v	04
05	GPIO03 (SCL1 , I²C)	Ground	06

The following command will display which devices are connected to the Pi.

```
        sudo i2cdetect -y 1
```

BMP180 Breakout is properly connected it should show up at 0x77 as follows:

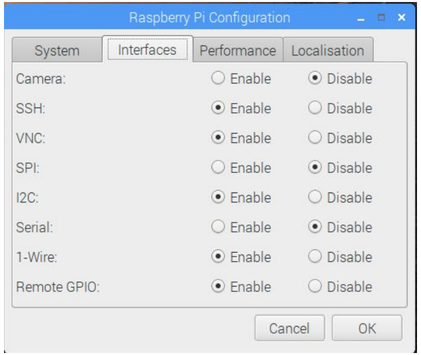


```
pi@walle-pi3-8gig:~ $ sudo i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- 77
```

Raspberry Pi BMP180 Software Setup

```
sudo apt-get update
sudo apt-get upgrade
```

Check that I2C is enabled using Raspberry Pi Configuration



Camera:	○ Enable	● Disable
SSH:	● Enable	○ Disable
VNC:	● Enable	○ Disable
SPI:	○ Enable	● Disable
I2C:	● Enable	○ Disable
Serial:	○ Enable	● Disable
1-Wire:	● Enable	○ Disable
Remote GPIO:	● Enable	○ Disable

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

Validate the device is attached correctly

sudo i2cdetect -y 1 (version 2 Raspberry Pi)

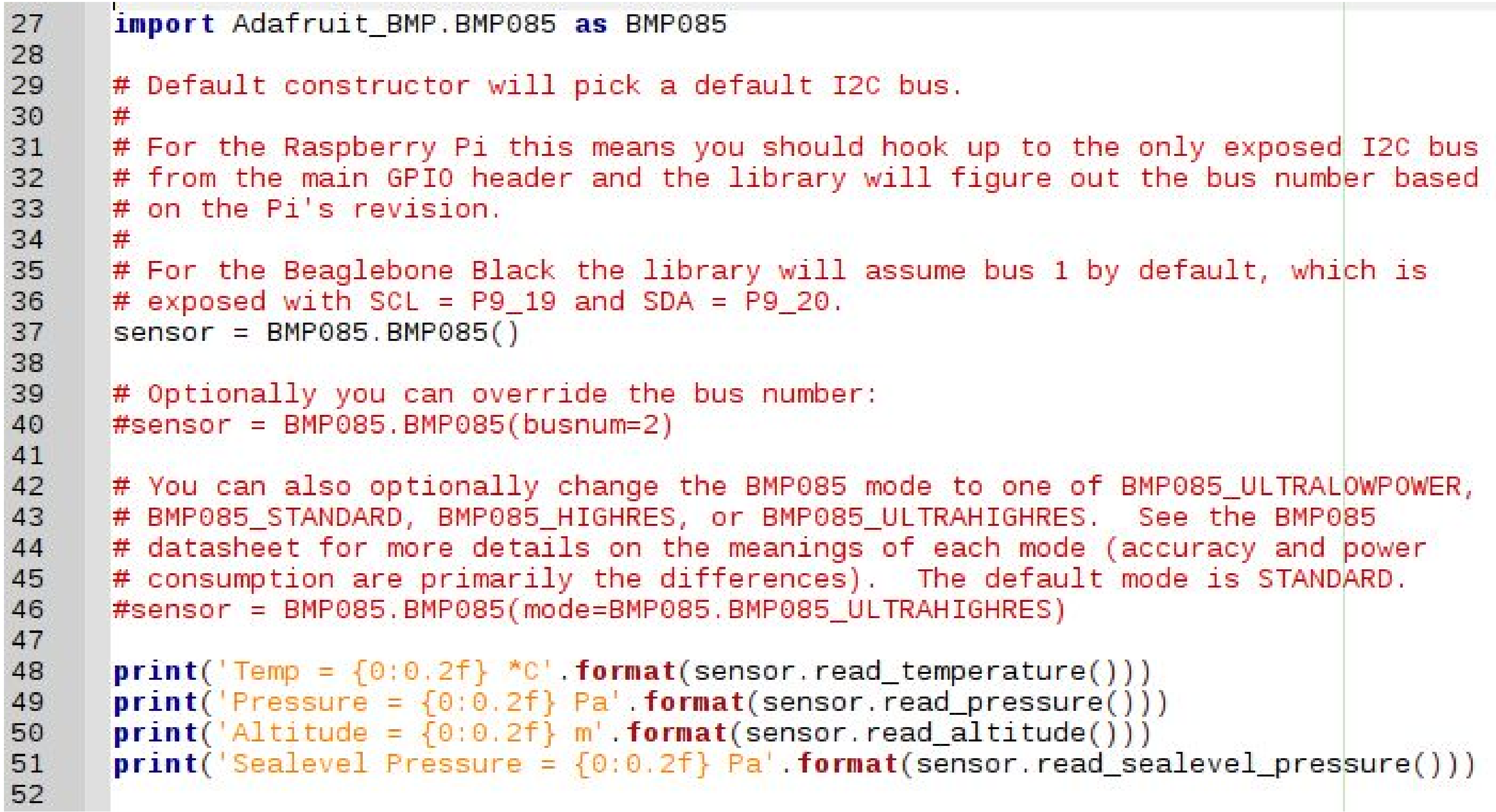
Install the software:

```
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

cd examples

sudo python simpletest.py
```

GitHub - Additional resources:

https://github.com/lexruee/bmp180

The Code: Simpletest.py



```python
import Adafruit_BMP.BMP085 as BMP085

# Default constructor will pick a default I2C bus.
#
# For the Raspberry Pi this means you should hook up to the only exposed I2C bus
# from the main GPIO header and the library will figure out the bus number based
# on the Pi's revision.
#
# For the Beaglebone Black the library will assume bus 1 by default, which is
# exposed with SCL = P9_19 and SDA = P9_20.
sensor = BMP085.BMP085()

# Optionally you can override the bus number:
#sensor = BMP085.BMP085(busnum=2)

# You can also optionally change the BMP085 mode to one of BMP085_ULTRALOWPOWER,
# BMP085_STANDARD, BMP085_HIGHRES, or BMP085_ULTRAHIGHRES.  See the BMP085
# datasheet for more details on the meanings of each mode (accuracy and power
# consumption are primarily the differences).  The default mode is STANDARD.
#sensor = BMP085.BMP085(mode=BMP085.BMP085_ULTRAHIGHRES)

print('Temp = {0:0.2f} *C'.format(sensor.read_temperature()))
print('Pressure = {0:0.2f} Pa'.format(sensor.read_pressure()))
print('Altitude = {0:0.2f} m'.format(sensor.read_altitude()))
print('Sealevel Pressure = {0:0.2f} Pa'.format(sensor.read_sealevel_pressure()))
```