Barometric Pressure Sensor: Bosch BMP280

Main Findings:

Senses:

- Barometric pressure
    - +/- 1 hPa absolute error
    - +/- 0.12 hPa relative accuracy
    - Resolution: 0.16 Pa
- Temperature
    - +/- 1.0 degree Celsius
    - Temperature Coefficient (TCO): 1.5 Pa/K
        - Temperature Drift: 12.6 cm/K
    - Resolution: 0.01 degree C
- Altitude
    - +/- 10m, if not calibrated with the pressure at sea level for the sensor's location and day
    - +/- 1m when calibrated
    - 0.25m is the maximum noise magnitude

Sample Rate: Minimum of 157 Hz

How It Works:

- Piezoresistive Pressure Sensor
    - Micro-electronic mechanical system on its own silicon chip
        - Strain gauge in Wheatstone Bridge, directly printed onto a diaphragm. The change in air pressure deflects the diaphragm and strains the sensors.

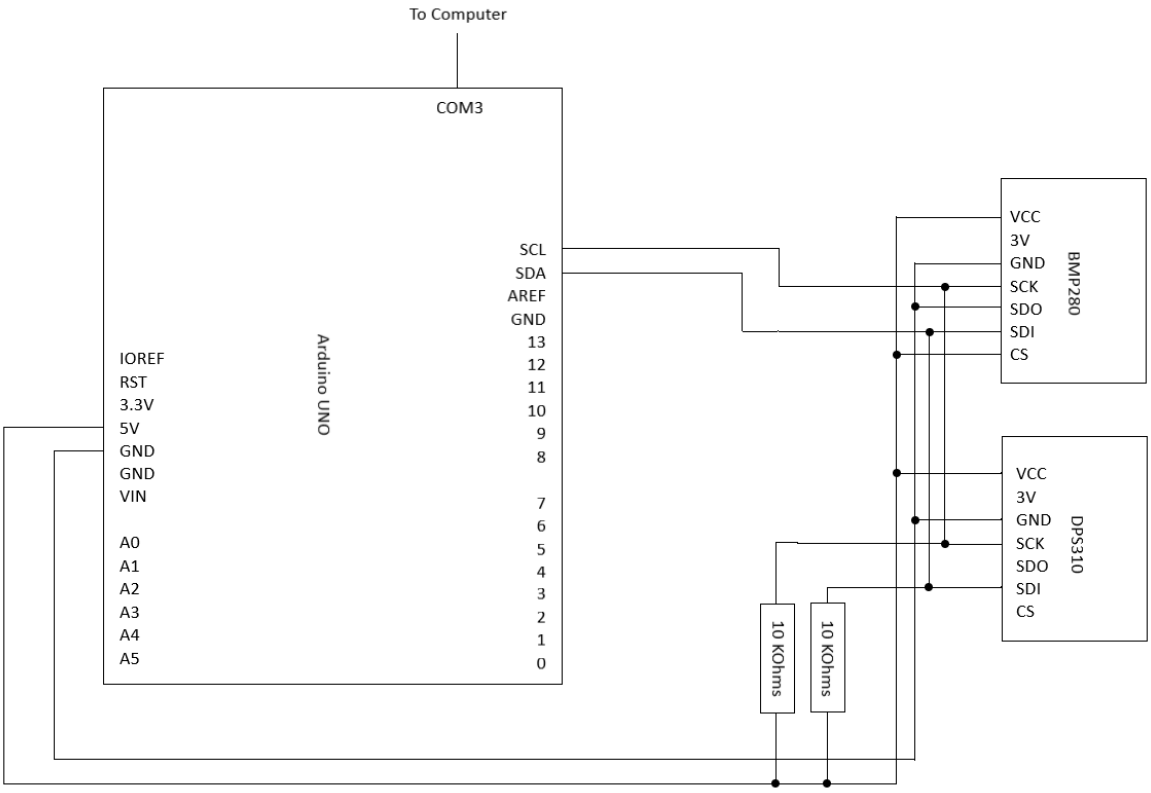
Pinout:

I2C or SPI wiring (I used I2C to reduce the amount of wires running to the Arduino)

Runs on 5V and 3.3V (here it's running on 5V because that's the Arduino's logic-level voltage)

- SCK – Clock pin
- SDI – Data pin
- VCC – 5V Power
- GND – Common ground for power and logic

Wiring Diagram:

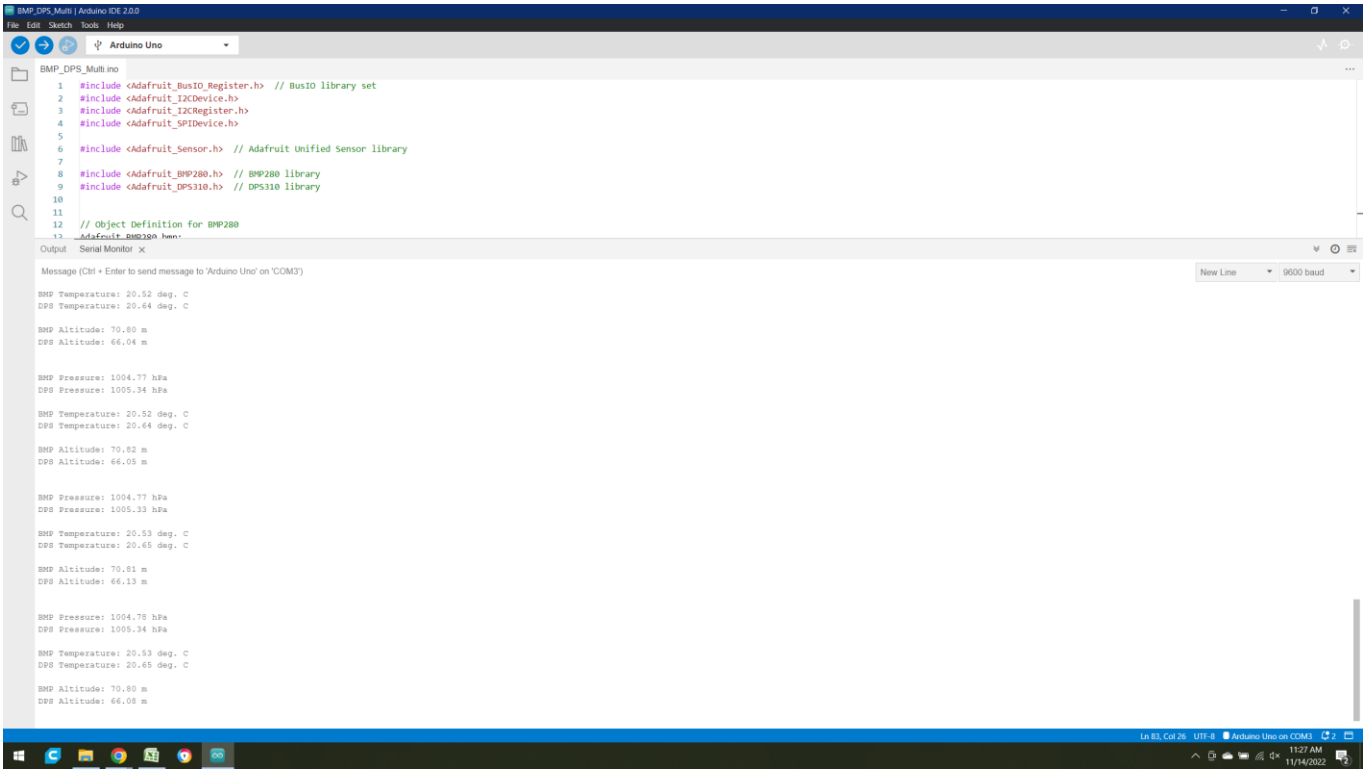

To Computer
COM3
Arduino UNO
IOREF
RST
3.3V
5V
GND
GND
VIN
A0
A1
A2
A3
A4
A5
SCL
SDA
AREF
GND
13
12
11
10
9
8
7
6
5
4
3
2
1
0
10 KOhms
10 KOhms
VCC
3V
GND
SCK
SDO
SDI
CS
BMP280
VCC
3V
GND
SCK
SDO
SDI
CS
DPS310

Arduino Code:

```
#include <Adafruit_BusIO_Register.h>  // BusIO library set
#include <Adafruit_I2CDevice.h>
#include <Adafruit_I2CRegister.h>
#include <Adafruit_SPIDevice.h>

#include <Adafruit_Sensor.h>  // Adafruit Unified Sensor library

#include <Adafruit_BMP280.h>  // BMP280 library
#include <Adafruit_DPS310.h>  // DPS310 library

// Object Definition for BMP280
Adafruit_BMP280 bmp;

// Object definition for DPS310
#define Adafruit
Adafruit_DPS310 dps;


void setup() {
 Serial.begin(9600); // Sets baud rate for both sensors

 // Checks to see if BMP280 is set up properly
 if(!bmp.begin(BMP280_ADDRESS_ALT, BMP280_CHIPID)) {
  Serial.println("BMP280 Error!");
  while (1);
 }

// Sampling modes for BMP280
 bmp.setSampling(Adafruit_BMP280::MODE_NORMAL,
             Adafruit_BMP280::SAMPLING_X2,
             Adafruit_BMP280::SAMPLING_X16,
             Adafruit_BMP280::FILTER_X16,
             Adafruit_BMP280::STANDBY_MS_500);

// Checks to see if DPS310 is set up properly
 if (!dps.begin_I2C()) {
  Serial.println("DPS310 Error!");
  while (1);
 }

// Sampling modes for DPS310
 dps.configurePressure(DPS310_64HZ, DPS310_64SAMPLES);
 dps.configureTemperature(DPS310_64HZ, DPS310_64SAMPLES);
}


void loop() {
 sensors_event_t temp_event, pressure_event; // Creates variables temp_event and pressure_event
 dps.getEvents(&temp_event, &pressure_event);  // Reads and assigns temp and presure to the variables
temp_event and presure_event
```

```
  float altitude_dps = dps.readAltitude(1013.25);

  float pressure = bmp.readPressure() / 100;  // Creates and assigns pressure variable in units of hPa
  float temperature = bmp.readTemperature();  // Creates and assigns temperature variable in deg. C
  float altitude = bmp.readAltitude(1013.25); // Creates and assigns altitude varaible in m

// Prints BMP280 pressure reading
  Serial.print("BMP Pressure: ");
  Serial.print(pressure);
  Serial.println(" hPa");

// Prints DPS310 pressure reading
  Serial.print("DPS Pressure: ");
  Serial.print(pressure_event.pressure);
  Serial.println(" hPa");
  Serial.println("");

// Prints BMP280 temoperature reading
  Serial.print("BMP Temperature: ");
  Serial.print(temperature);
  Serial.println(" deg. C");

// Prints DPS310 temperature reading
  Serial.print("DPS Temperature: ");
  Serial.print(temp_event.temperature);
  Serial.println(" deg. C");
  Serial.println("");

// Prints BMP280 altitude reading
  Serial.print("BMP Altitude: ");
  Serial.print(altitude);
  Serial.println(" m");

// Prints DPS310 altitude reading
  Serial.print("DPS Altitude: ");
  Serial.print(altitude_dps);
  Serial.println(" m");

  Serial.println("");
  Serial.println("");

  delay(1000);  // Delays one second, so the chips' oversampling process has time to occur
}
```

Sensor Readings:



Selected Characteristics:

To test these characteristics, I need to control the temperature, pressure, and altitude experienced by the sensor. For pressure, I will use a bike pump to pressurize a sealed chamber with the sensor inside. For temperature, I will cool the sensor by placing the pressure chamber in a bowl of ice water, and I will heat the sensor by blowing it indirectly with a hair dryer. For altitude, I will collect data while riding my apartment building's elevator from the ground floor to the fourth floor. I will compare the BMP data to the DPS310 chip, which is more accurate and has a smaller resolution, in order to have a ground truth. The DPS will be placed next to the BMP chip during measurements.

Characteristic

- Resolution
    - By squeezing the pressure chamber when sealed or lifting the sensor slowly and recording when the reading changes, the resolution of the sensor can be found.
- Accuracy
    - The sensor will detect the pressure/temperature/altitude of the chamber across the temperature/pressure/altitude range. By comparing the readings to the DPS310 sensor, which has a better resolution than the BMP280, the accuracy of the sensor over the range can be detected.
- Drift
    - The sensor will be left at different pressures/temperatures/altitudes for a long period of time. Its variance is recorded as the sensor drift.
- Range
    - The sensor readings will be brought up to the upper limit of what the BMP is rated to endure and brought as low as I can get with the resources I have. For pressure, in the chamber with the bike pump, and for temperature, the hair dryer and ice bath.
- Linearity
    - The pressure/temperature/altitude of the chamber is increased or decreased, and the sensor readings at those levels are plotted. The data will be linearly regressed, and the fit of the curve to the data provides insight into how linear the sensor is.
- Sensitivity
    - This is the slope of the linearly regressed dataset.

Data Sheet:

1. Specification

If not stated otherwise,

- All values are valid over the full voltage range
- All minimum/maximum values are given for the full accuracy temperature range
- Minimum/maximum values of drifts, offsets and temperature coefficients are ±3σ values over lifetime
- Typical values of currents and state machine timings are determined at 25 °C
- Minimum/maximum values of currents are determined using corner lots over complete temperature range
- Minimum/maximum values of state machine timings are determined using corner lots over 0…+65 °C temperature range

The specification tables are split into pressure and temperature part of BMP280

Table 2: Parameter specification

Parameter	Symbol	Condition	Min	Typ	Max	Units
Operating temperature range	T_A	operational	-40	25	+85	°C
		full accuracy	0		+65	
Operating pressure range	P	full accuracy	300		1100	hPa
Sensor supply voltage	V_{DD}	ripple max. 50mVpp	1.71	1.8	3.6	V
Interface supply voltage	V_{DDIO}		1.2	1.8	3.6	V
Supply current	$I_{DD,LP}$	1 Hz forced mode, pressure and temperature, lowest power		2.8	4.2	µA
Peak current	I_{peak}	during pressure measurement		720	1120	µA
Current at temperature measurement	I_{DDT}			325		µA
Sleep current[1]	I_{DDSL}	25 °C		0.1	0.3	µA
Standby current (inactive period of normal mode) [2]	I_{DDSB}	25 °C		0.2	0.5	µA
Relative accuracy pressure V_{DD} = 3.3V	A_{rel}	700 … 900hPa		±0.12		hPa
		25 . . . 40 °C		±1.0		m

[1] Typical value at VDD = VDDIO = 1.8 V, maximal value at VDD = VDDIO = 3.6 V.
[2] Typical value at VDD = VDDIO = 1.8 V, maximal value at VDD = VDDIO = 3.6 V.

Offset temperature coefficient	TCO	900hPa		±1.5		Pa/K
		25 . . . 40 °C		12.6		cm/K
Absolute accuracy pressure	A^P_{ext}	300 . . . 1100 hPa -20 . . . 0 °C		±1.7		hPa
	A^P_{full}	300 . . . 1100 hPa 0 . . . 65 °C		±1.0		hPa
Resolution of output data in ultra high resolution mode	R^P	Pressure		0.0016		hPa
	R^T	Temperature		0.01		°C
Noise in pressure	$V_{p,full}$	Full bandwidth, ultra high resolution See chapter 3.5		1.3		Pa
				11		cm
	$V_{p,filtered}$	Lowest bandwidth, ultra high resolution See chapter 3.5		0.2		Pa
				1.7		cm
Absolute accuracy temperature[3]	A^T	@ 25 °C		±0.5		°C
		0 . . . +65 °C		±1.0		°C
PSRR (DC)	PSRR	full V_{DD} range			±0.005	Pa/mV
Long term stability[4]	ΔP_{stab}	12 months		±1.0		hPa
Solder drifts		Minimum solder height 50 µm	-0.5		+2	hPa
Start-up time	$t_{startup}$	Time to first communication after both V_{DD} > 1.58V and V_{DDIO} > 0.65V			2	ms
Possible sampling rate	f_{sample}	*osrs_t* = *osrs_p* = 1; See chapter 3.8	157	182	tbd[5]	Hz
Standby time accuracy	$\Delta t_{standby}$			±5	±25	%

[3] Temperature measured by the internal temperature sensor. This temperature value depends on the PCB temperature, sensor element self-heating and ambient temperature and is typically above ambient temperature.
[4] Long term stability is specified in the full accuracy operating pressure range 0 ... 65°C
[5] Depends on application case, please contact Application Engineer for further questions

2. Absolute maximum ratings

The absolute maximum ratings are provided in Table 3.

Table 3: Absolute maximum ratings

Parameter	Condition	Min	Max	Unit
Voltage at any supply pin	V_{DD} and V_{DDIO} Pin	-0.3	4.25	V
Voltage at any interface pin		-0.3	V_{DDIO} + 0.3	V
Storage Temperature	≤ 65% rel. H.	-45	+85	°C
Pressure		0	20 000	hPa
ESD	HBM, at any Pin		±2	kV
	CDM		±500	V
	Machine model		±200	V

Sources:

Sources:

https://cdn-learn.adafruit.com/downloads/pdf/adafruit-bmp280-barometric-pressure-plus-temperature-sensor-breakout.pdf

https://simple-circuit.com/arduino-bmp280-sensor-lcd/

http://wiki.sunfounder.cc/index.php?title=BMP280_Pressure_Sensor_Module

https://www.bosch-sensortec.com/products/environmental-sensors/pressure-sensors/bmp280/

http://wiki.sunfounder.cc/images/d/d1/BMP280_datasheet.pdf

https://www.avnet.com/wps/portal/abacus/solutions/technologies/sensors/pressure-sensors/core-technologies/piezoresistive-strain-gauge/

https://www.avnet.com/wps/portal/abacus/solutions/technologies/sensors/pressure-sensors/core-technologies/mems/