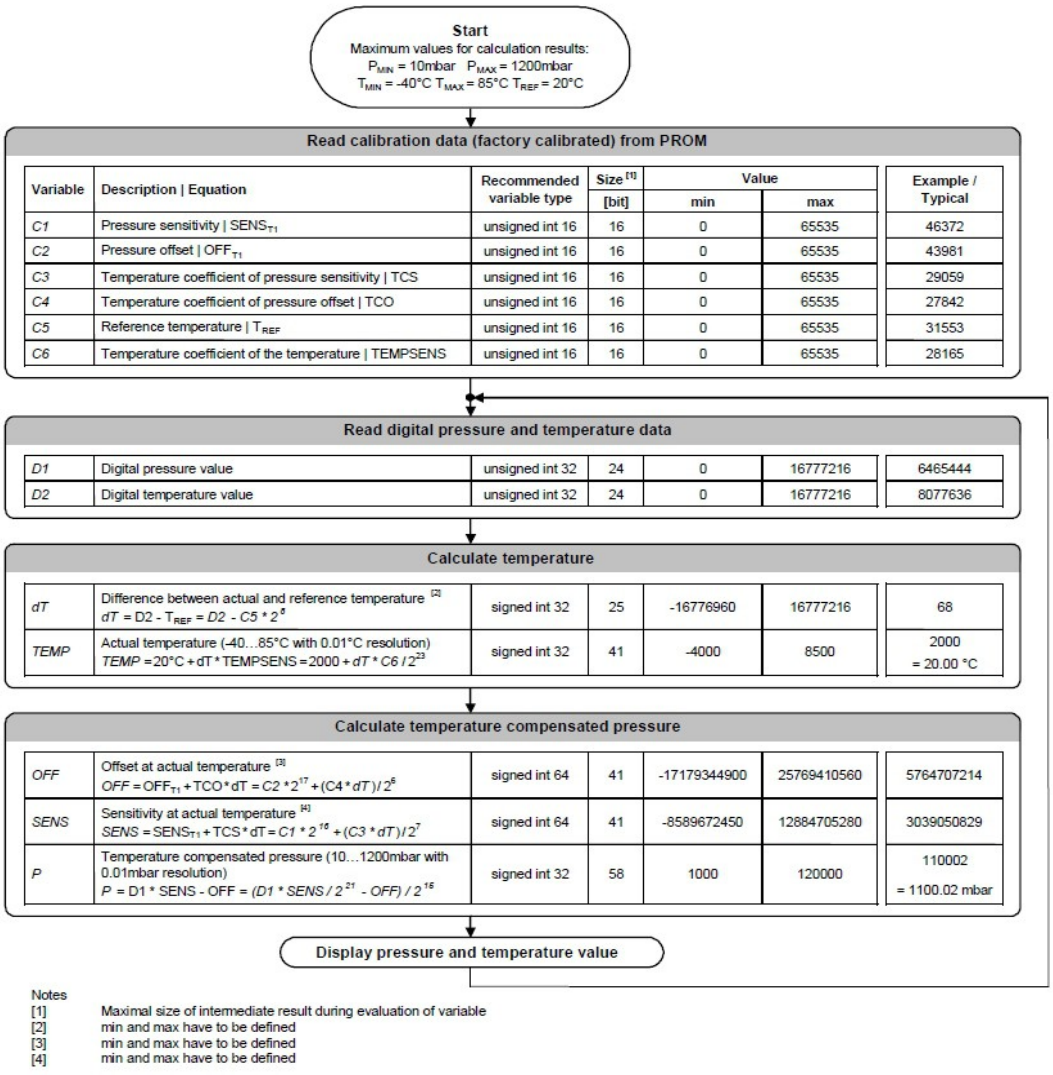
This document describes the calibration method devised by David Edwards for performing a first order calibration of the MS5607-02BA03 Barometric Pressure Sensor on an Arduino processor, which is limited to 32 bit integer operations.

The MS5607 contains factory calibration constants, which must be used to calibrate the raw temperature and pressure data, as described in the MS5607 specifications:

FACTORY CALIBRATION

Every module is individually factory calibrated at two temperatures and two pressures. As a result, 6 coefficients necessary to compensate for process variations and temperature variations are calculated and stored in the 128-bit PROM of each module. These bits (partitioned into 6 coefficients) must be read by the microcontroller software and used in the program converting D1 and D2 into compensated pressure and temperature values.

The calculations involve operations on signed and unsigned integers with a maximum size of 64 bits. The standard Arduino Long integers have only 32 bit capacity, so if the algorithm specified in Fig 2 above is used, the calculations are inaccurate because of over and under flows.



Variable	Description | Equation	Recommended variable type	Size [1] [bit]	Value min	Value max	Example / Typical
C1	Pressure sensitivity | $SENS_{T1}$	unsigned int 16	16	0	65535	46372
C2	Pressure offset | OFF_{T1}	unsigned int 16	16	0	65535	43981
C3	Temperature coefficient of pressure sensitivity | TCS	unsigned int 16	16	0	65535	29059
C4	Temperature coefficient of pressure offset | TCO	unsigned int 16	16	0	65535	27842
C5	Reference temperature | T_{REF}	unsigned int 16	16	0	65535	31553
C6	Temperature coefficient of the temperature | TEMPSENS	unsigned int 16	16	0	65535	28165
D1	Digital pressure value	unsigned int 32	24	0	16777216	6465444
D2	Digital temperature value	unsigned int 32	24	0	16777216	8077636
dT	Difference between actual and reference temperature [2] $dT = D2 - T_{REF} = D2 - C5 * 2^8$	signed int 32	25	-16776960	16777216	68
TEMP	Actual temperature (-40…85°C with 0.01°C resolution) $TEMP = 20°C + dT * TEMPSENS = 2000 + dT * C6 / 2^{23}$	signed int 32	41	-4000	8500	2000 = 20.00 °C
OFF	Offset at actual temperature [3] $OFF = OFF_{T1} + TCO * dT = C2 * 2^{17} + (C4 * dT) / 2^6$	signed int 64	41	-17179344900	25769410560	5764707214
SENS	Sensitivity at actual temperature [4] $SENS = SENS_{T1} + TCS * dT = C1 * 2^{16} + (C3 * dT) / 2^7$	signed int 64	41	-8589672450	12884705280	3039050829
P	Temperature compensated pressure (10…1200mbar with 0.01mbar resolution) $P = D1 * SENS - OFF = (D1 * SENS / 2^{21} - OFF) / 2^{15}$	signed int 32	58	1000	120000	110002 = 1100.02 mbar

Notes
[1] Maximal size of intermediate result during evaluation of variable
[2] min and max have to be defined
[3] min and max have to be defined
[4] min and max have to be defined

Figure 2: Flow chart for pressure and temperature reading and software compensation.

This is because the algorithm requires scaling by factors of up to 2^23.

The first order calibration uses the following algorithm:

$$P = (D1 \times SENS / 2097152 - OFF) / 65536$$

where:

P = Pressure in Pa
D1 is the raw pressure data
SENS and OFF are the calculated sensitivity (gain) and offset calculated from the calibration coefficients and the raw temperature reading.

SENS and OFF are calculated as follows:

$$SENS = C1 \times 65536 + C3 \times DT / 128$$

$$OFF = C2 \times 131072 + C4 \times DT / 64$$

where:

C1, C2, C3 & C4 are factory calibration coefficients read from the MS5607 sensor
DT is the difference difference

DT is calculated as follows:

$$DT = D2 - C5 \times 256$$

where:

D2 is the raw temperature data
C5 is a factory calibration coefficient read from the MS5607 sensor

The MS5607 temperature is not needed to calculate pressure, but can be calculated as follows:

$$TEMP = 2000 + DT \times C6 / 8388608$$

where:

TEMP is the calibrated temperature in degC, multiplied by 100
DT is the temperature difference
C6 is a factory calibration coefficient read from the MS5607 sensor

By algebraically expanding these equations, pressure P can be calculated as:

$$P = P1 + P2 – P3 – P4$$

where:

$$P1 = (D1 / 1024) \times C1 / 1024)$$

$$P2 = (D1 / 16384) \times (C3 / 16384) \times (DT / 32768)$$

$$P3 = C2 \times 4$$

$$P4 = (C4 / 1024) \times (DT / 2048)$$

Each of the bracketed terms is evaluated before the terms are multiplied.

To obtain air pressure in hPa or mBars, a further division by 100 is performed.

Arduino code to perform these calculations is shown below. Note that nSamples are taken, calibrated and then averaged:

```
// Read pressure and temperature
TempAccum = 0;
PressAccum = 0;
for(i=0; i<nSamples; i++)
{
  // read temperature
  RawTemp = ReadADC(0x18);   // typical value 8077636
  DeltaTemp = RawTemp - coefficients_[4] * 256;
  CalTemp = 2000 + (DeltaTemp * coefficients_[5] >> 23);
  TempAccum += CalTemp;
  // read pressure
  RawPress = ReadADC(0x08);  // typical value 6465444
  P1 = (float(RawPress)/1024L)*(float(coefficients_[0])/1024L);
  P2 = (float(RawPress)/16384L)*(float(coefficients_[2])/16384L)*(float(DeltaTemp)/32768L);
  P3 = (float(coefficients_[1])*4);
  P4 = (float(coefficients_[3])/1024l)*(float(DeltaTemp)/2048L);
  PressAccum += P1 + P2 - P3 - P4;
}
Temp = TempAccum/nSamples/100;
Press = PressAccum/nSamples/100;
```

Arduino code to read the calibration coefficients is shown below:

```
  unsigned int ReadCoefficient(const byte coefNum)
  {
      unsigned int rC=0;
      Wire.beginTransmission(i2cAddr_);
      Wire.write(cmdPromRd_ + coefNum * 2); // send PROM READ command
      Wire.endTransmission();
      const byte q2 = 2;
      Wire.requestFrom(i2cAddr_, q2);
      if(Wire.available() >= 2)
      {
          unsigned int ret = Wire.read();   // read MSB
          unsigned int rC  = 256 * ret;
          ret = Wire.read();                // read LSB
          rC  = rC + ret;
          return rC;
      }
      return 0;
  }
```

Note that Cofficient C1 is stored in the coefficients_[0] array element.

The spreadsheet extract below shows a typical calibration:

Parameter	Variable	Name	Value
Coefficient 0	C1	SENS	46372
Coefficient 1	C2	OFF	43981
Coefficient 2	C3	TCS	29059
Coefficient 3	C4	TCO	27842
Coefficient 4	C5	Tref	31553
Coefficient 5	C6	TempSens	28165

Parameter	Variable	Value	Units
Raw Temp	D2	8077636	
Raw Press	D1	6465444	
DeltaTemp	DT	68	
Temperature	TEMP	2000.23	
Temperature	TEMP	20.00	DegC
Pressure Offset	OFF	5764707214	
Pressure Sensitivity	SENS	3039050830	
Pressure	P	110002.96	Pa
Pressure	P	1100.03	hPa

D1/2^10)	6313.9101563	
C1/2^10	45.28515625	
	P1	285926.40797
D1/2^14	394.61938477	
C3/2^14	1.7736206055	
DT/2^15	0.0020751953	
	P2	1.4524397249
C2*2^2	175924	
	P3	-175924
C4/2^10	27.189453125	
DT/2^11	0.033203125	
	P4	-0.902774811
	P	110002.95764
	P/100	1100.03

The upper block shows the temperature and pressure calculations performed using the MS5607 calculation method, which gives the correct answers because of the high precision spreadsheet calculations.

The lower block shows the pressure calculations performed as per the method outlined above.

David Edwards
8 August 2014