SENSOR SIGNAL CONDITIONERS TEMPERATURE SENSORS and THERMAL MANAGEMENT

Keywords: Compensation algorithm, Multi-channel signal conditioner, digital signal conditioner, High performance signal conditioner, Low cost signal conditioner, Precision Sensor Signal Conditioner, Pressure sensor compensation, PRT sensor compensation, Thick

Sep 23, 2005

APPLICATION NOTE 3649

MAX1464 Signal-Conditioner, Sensor Compensation Algorithm

The MAX1464 is a fully digital, high-performance signal conditioner with multichannel inputs, analog and digital outputs, and support for 4-20mA output applications. The MAX1464 can be programmed to correct the nonlinearities and temperature dependent characteristics of sensors.

This document describes the procedures to compensate and calibrate a sensor signal that is applied to one of the MAX1464's ADC channels. The algorithm provides offset and span correction over the specified temperature range. It is assumed that the MAX1464 VDD supply voltage does not change during the compensation process and operation of the device.

The data presented in this document is real data acquired from a piezoresistive pressure sensor used to demonstrate the compensation algorithm.

1. Input Parameters

The user must define some input parameters for the application. They are:

Tmin = minimum temperature, in degrees Celsius
Tintr = intermediate temperature, in degrees Celsius
Tmax = maximum temperature, in degrees Celsius
Pmin = minimum pressure
Pmax = maximum pressure
Vmin = desired MAX1464 output, at the minimum pressure, in volts
Vmax = desired MAX1464 output, at the maximum pressure, in volts

The next limits are application dependent, and can differ for other applications:

$Tmin := -40$ $\quad Pmin := 0$ $\quad Vmin := 0.5$

$Tintr := 25$ $\quad Pmax := 15$ $\quad Vmax := 4.5$

$Tmax := 125$

For pressure nonlinearity correction, we define Pmed as the sensor excitation midpoint:

$$Pmed := \frac{Pmax - Pmin}{2} + Pmin$$

$$Pmed = 7.5$$

For better performance and to maximize the range of the MAX1464's ADC converter, you must adjust the appropriate coarse offset and PGA settings. Select the temperature for which the sensor sensitivity is the highest, and apply minimum and maximum sensor excitation. The user should then select the PGA gain and coarse offset settings that maximize the ADC output at these conditions.

The ADC acquired data must then be entered in the matrix below. Each row of data has the normalized ADC reading for the minimum, medium, and maximum sensor excitation at the indicated temperature. Each row also has the ADC reading for the MAX1464's internal temperature sensor, and the DAC output voltage (through either the small or large op amp) for a fixed, normalized digital input of -0.5 (DACinM) and +0.5 (DACinP).

temperature := 4 (column index of temperature data)

DACinM := −0.5

DACinP := 0.5

The acquired data (ad) matrix is shown below, with the ADC results entered in hexadecimal, and the DAC output voltage readings entered in decimal. Due to software limitations, hexadecimal values must be entered with a leading zero (0).

$$ad := \begin{pmatrix} 0 & Pmin & Pmed & Pmax & Temperature & DACinM & DACinP \\ Tmin & 0A836h & 00741h & 0659Fh & 0E0AFh & 1.0040 & 3.9855 \\ Tintr & 0B148h & 001EAh & 05229h & 0F8C7h & 1.0047 & 3.9867 \\ Tmax & 0B6B0h & 0F6F5h & 03713h & 01C9Fh & 1.0053 & 3.9885 \end{pmatrix}$$

To convert the two's complement hexadecimal values into decimal values (between -1 and +1), the following function is defined:

$$h2d(x) := if\left(x \geq 2^{15}, \frac{x - 2^{16}}{2^{15}}, \frac{x}{2^{15}}\right)$$

The decimal representation of the acquired data matrix is then defined as:

$$data := \begin{pmatrix} 0 & Pmin & Pmed & Pmax & Temperature & DACinM & DACinP \\ Tmin & h2d(ad_{1,1}) & h2d(ad_{1,2}) & h2d(ad_{1,3}) & h2d(ad_{1,4}) & ad_{1,5} & ad_{1,6} \\ Tintr & h2d(ad_{2,1}) & h2d(ad_{2,2}) & h2d(ad_{2,3}) & h2d(ad_{2,4}) & ad_{2,5} & ad_{2,6} \\ Tmax & h2d(ad_{3,1}) & h2d(ad_{3,2}) & h2d(ad_{3,3}) & h2d(ad_{3,4}) & ad_{3,5} & ad_{3,6} \end{pmatrix}$$

With the defined user values, the above matrix is shown as:

$$data = \begin{pmatrix} 0 & 0 & 7.5 & 15 & 4 & -0.5 & 0.5 \\ -40 & -0.68585205 & 0.05667114 & 0.79391479 & -0.24465942 & 1.004 & 3.9855 \\ 25 & -0.61499023 & 0.01495361 & 0.64187622 & -0.056427 & 1.0047 & 3.9867 \\ 125 & -0.57275391 & -0.07064819 & 0.43026733 & 0.22360229 & 1.0053 & 3.9885 \end{pmatrix}$$

Other parameters used throughout the document are defined as:

$$Tmed := \frac{Tmax - Tmin}{2} + Tmin$$

$$Tint1 := \frac{Tmax - Tmin}{3} + Tmin$$

$$Tint2 := Tmax - \frac{Tmax - Tmin}{3}$$

$$T := Tmin, Tmin + 1 .. Tmax$$

$$P := Pmin, Pmin + \frac{Pmax - Pmin}{(Pmax - Pmin) \cdot 10} .. Pmax$$

$$F2(x) := \begin{pmatrix} 1 \\ x \\ x^2 \end{pmatrix} \qquad F3(x) := \begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \end{pmatrix} \qquad F4(x) := \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \end{bmatrix}$$

$tmin := 1$ $\quad pmin := 1$ $\quad inm := 5$
$tint := 2$ $\quad pmed := 2$ $\quad inp := 6$
$tmax := 3$ $\quad pmax := 3$

$$Ptmin := \begin{pmatrix} data_{tmin,pmin} \\ data_{tmin,pmed} \\ data_{tmin,pmax} \end{pmatrix} \qquad Temp := \begin{pmatrix} data_{tmin,temperature} \\ data_{tint,temperature} \\ data_{tmax,temperature} \end{pmatrix}$$

$$Ptint := \begin{pmatrix} data_{tint,pmin} \\ data_{tint,pmed} \\ data_{tint,pmax} \end{pmatrix} \qquad DacM := \begin{pmatrix} data_{tmin,inm} \\ data_{tint,inm} \\ data_{tmax,inm} \end{pmatrix}$$

$$Ptmax := \begin{pmatrix} data_{tmax,pmin} \\ data_{tmax,pmed} \\ data_{tmax,pmax} \end{pmatrix} \qquad DacP := \begin{pmatrix} data_{tmin,inp} \\ data_{tint,inp} \\ data_{tmax,inp} \end{pmatrix}$$

2. Data Modeling

This section shows the mathematical data modeling of the sensor, temperature, and DAC data. The derived functions for the sensor, temperature, and DAC will then be used as their model.

2.1. Sensor Data Modeling

We first model the sensor data at each individual temperature, and we then model the variation of the coefficients over temperature.

The coefficients are found solving a linear system of equations, described as:

$A\,x = b$
$x = A^{-1}\,b$

where "A" is a square matrix, and "x" and "b" are column vectors:

The inverse matrix, in this case, is defined as:

$$Pinv := \begin{pmatrix} 1 & Pmin & Pmin^2 \\ 1 & Pmed & Pmed^2 \\ 1 & Pmax & Pmax^2 \end{pmatrix}^{-1}$$

For the minimum temperature, the second-order coefficients that model the variation of the ADC data over pressure are given by:

$$a := Pinv \cdot Ptmin$$

$$a = \begin{pmatrix} -0.68585205 \\ 0.09935506 \\ -4.69292535 \cdot 10^{-5} \end{pmatrix}$$

The equation that models the ADC output over pressure at this temperature is given by:

$$P0(P) := a_0 + a_1 \cdot P + a_2 \cdot P^2$$

For the intermediate temperature, the second-order coefficients that model the variation of the ADC data over pressure are given by:

$$b := Pinv \cdot Ptint$$

$$b = \begin{pmatrix} -0.61499023 \\ 0.08419393 \\ -2.68554687 \cdot 10^{-5} \end{pmatrix}$$

The equation that models the ADC output over pressure at this temperature is given by:

$$P1(P) := b_0 + b_1 \cdot P + b_2 \cdot P^2$$

For the maximum temperature, the second-order coefficients that model the variation of the ADC data over pressure are given by:

$$c := Pinv \cdot Ptmax$$

$$c = \begin{pmatrix} -0.57275391 \\ 0.06702677 \\ -1.05794271 \cdot 10^{-5} \end{pmatrix}$$

The equation that models the ADC output over pressure at this temperature is given by:

$$P2(P) := c_0 + c_1 \cdot P + c_2 \cdot P^2$$

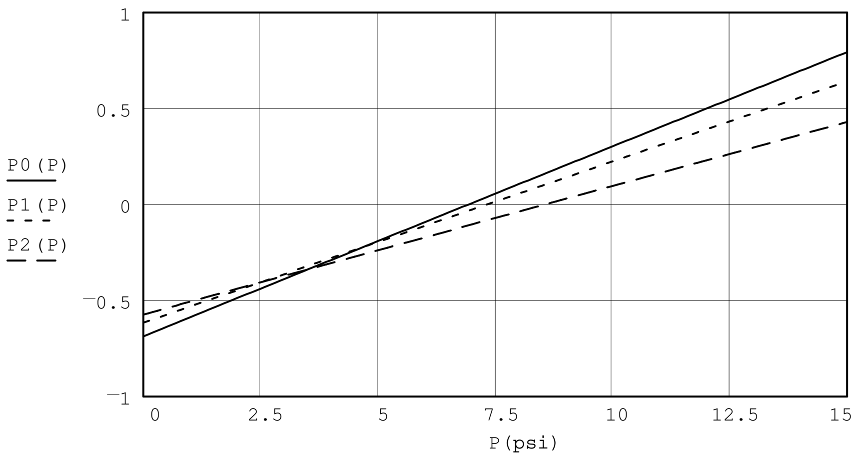

Figure 1. ADC output as a function of sensor excitation.

Now, we combine these equations to include the temperature dependency. Basically, we will find second-order equations that model the coefficients' variation over temperature. The zero-, first-, and second-order coefficients on P0(T), P1(T), and P2(T) are given by:

$$P0 := \begin{pmatrix} a_0 \\ b_0 \\ c_0 \end{pmatrix} \qquad P1 := \begin{pmatrix} a_1 \\ b_1 \\ c_1 \end{pmatrix} \qquad P2 := \begin{pmatrix} a_2 \\ b_2 \\ c_2 \end{pmatrix}$$

For the temperature modeling of the coefficients, we now need to define the following matrix:

$$\mathrm{Tinv} := \begin{pmatrix} 1 & \mathrm{Tmin} & \mathrm{Tmin}^2 \\ 1 & \mathrm{Tintr} & \mathrm{Tintr}^2 \\ 1 & \mathrm{Tmax} & \mathrm{Tmax}^2 \end{pmatrix}^{-1}$$

The zero-order coefficients' dependency over temperature can be found as follows:

$$d := \mathrm{Tinv} \cdot p0$$

$$d = \begin{pmatrix} -0.63819739 \\ 0.00102947 \\ -4.04738491 \cdot 10^{-6} \end{pmatrix}$$

The zero-order coefficient function is given by:

$$C0(T) := d_0 + d_1 \cdot T + d_2 \cdot T^2$$

The first-order coefficients' dependency over temperature can be found as follows:

$$e := \mathrm{Tinv} \cdot p1$$

$$e = \begin{pmatrix} 0.08965194 \\ -2.2765032 \cdot 10^{-4} \\ 3.73191804 \cdot 10^{-7} \end{pmatrix}$$

The first-order coefficient function is given by:

$$C1(T) := e_0 + e_1 \cdot T + e_2 \cdot T^2$$

The second-order coefficients' dependency over temperature can be found as follows:

$$f := \mathrm{Tinv} \cdot p2$$

$$f = \begin{pmatrix} -3.36909004 \cdot 10^{-5} \\ 2.95548635 \cdot 10^{-7} \\ -8.85254792 \cdot 10^{-10} \end{pmatrix}$$

The second-order coefficient function is given by:

$$C2(T) := f_0 + f_1 \cdot T + f_2 \cdot T^2$$

The ADC output, as a function of both temperature and pressure, is then given by:

$$Pdata(T,P) := C0(T) + C1(T) \cdot P + C2(T) \cdot P^2$$

To verify the validity of the above equation, we compare the data matrix with the values from the Pdata function.

$$data = \begin{pmatrix} 0 & 0 & 7.5 & 15 & 4 & -0.5 & 0.5 \\ -40 & -0.68585205 & 0.05667114 & 0.79391479 & -0.24465942 & 1.004 & 3.9855 \\ 25 & -0.61499023 & 0.01495361 & 0.64187622 & -0.056427 & 1.0047 & 3.9867 \\ 125 & -0.57275391 & -0.07064819 & 0.43026733 & 0.22360229 & 1.0053 & 3.9885 \end{pmatrix}$$

$Pdata(-40,0) = -0.68585205$
$Pdata(25,0) = -0.61499023$
$Pdata(125,0) = -0.57275391$

$Pdata(-40,7.5) = 0.05667114$

$Pdata(25,7.5) = 0.01495361$

$Pdata(125,7.5) = -0.07064819$

$Pdata(-40,15) = 0.79391479$
$Pdata(25,15) = 0.64187622$
$Pdata(125,15) = 0.43026733$

2.2. Temperature Sensor Data Modeling

The MAX1464's internal temperature sensor must also be modeled. The ADC temperature data was previously defined, and is given below:

$$Temp = \begin{pmatrix} -0.24465942 \\ -0.056427 \\ 0.22360229 \end{pmatrix}$$

The second-order temperature coefficients are then given by:

$$t := Tinv \cdot Temp$$

$$t = \begin{pmatrix} -0.12824475 \\ 0.00288719 \\ -5.79336029 \cdot 10^{-7} \end{pmatrix}$$

The ADC output, as a function of temperature (**Figure 2**) is then given by:

$$Tdata(T) := t_0 + t_1 \cdot T + t_2 \cdot T^2$$

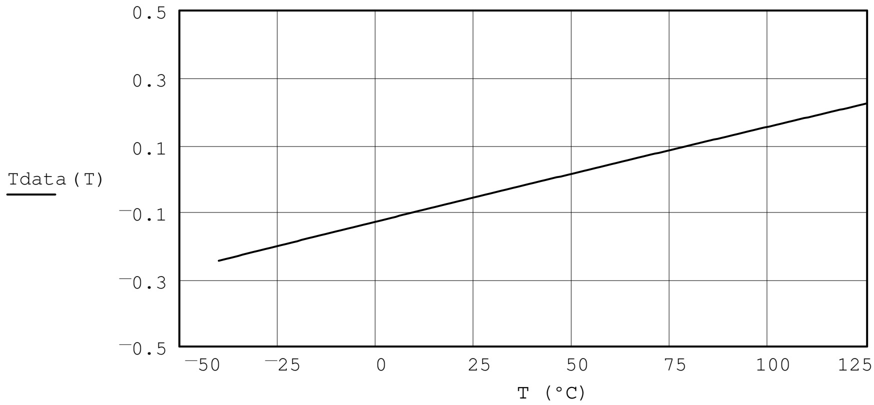

Figure 2. ADC temperature output as a function of temperature.

For verification, the values below show the input temperature data, and the values obtained from the Tdata function.

$$Temp = \begin{pmatrix} -0.24465942 \\ -0.056427 \\ 0.22360229 \end{pmatrix} \qquad \begin{aligned} Tdata(Tmin) &= -0.24465942 \\ Tdata(Tintr) &= -0.056427 \\ Tdata(Tmax) &= 0.22360229 \end{aligned}$$

2.3. DAC Data Modeling

The MAX1464's DAC must also be modeled to properly adjust its input values to the variations over temperature and process (gain, offset). The DAC data was already defined, and is given below for both the minus input (-0.5) and the positive input (+0.5):

$$DacM = \begin{pmatrix} 1.004 \\ 1.0047 \\ 1.0053 \end{pmatrix} \qquad DacP = \begin{pmatrix} 3.9855 \\ 3.9867 \\ 3.9885 \end{pmatrix}$$

The DAC gains for the input measured values are defined as:

$$DacGain := \begin{bmatrix} \dfrac{DacP_0 - DacM_0}{DACinP - DACinM} \\ \dfrac{DacP_1 - DacM_1}{DACinP - DACinM} \\ \dfrac{DacP_2 - DacM_2}{DACinP - DACinM} \end{bmatrix} \qquad DacGain = \begin{pmatrix} 2.9815 \\ 2.982 \\ 2.9832 \end{pmatrix}$$

The DAC offsets for the input measured values are defined as:

$$DacOffset := \begin{pmatrix} DacP_0 - DacGain_0 \cdot DACinP \\ DacP_1 - DacGain_1 \cdot DACinP \\ DacP_2 - DacGain_2 \cdot DACinP \end{pmatrix} \qquad DacOffset = \begin{pmatrix} 2.49475 \\ 2.4957 \\ 2.4969 \end{pmatrix}$$

The coefficients of the second-order function that represents the DAC gain over temperature are:

$$g := Tinv \cdot DacGain$$

$$g = \begin{pmatrix} 2.98178159 \\ 8.08391608 \cdot 10^{-6} \\ 2.61072261 \cdot 10^{-8} \end{pmatrix}$$

The DAC gain function is then given by:

$$dacgain(T) := g_0 + g_1 \cdot T + g_2 \cdot T^2$$

The coefficients of the second-order function that represents the DAC offset over temperature are:

$$h := Tinv \cdot DacOffset$$

$$h = \begin{pmatrix} 2.49535047 \\ 1.43776224 \cdot 10^{-5} \\ -1.58508159 \cdot 10^{-8} \end{pmatrix}$$

The DAC offset function is then given by:

$$dacoffset(T) := h_0 + h_1 \cdot T + h_2 \cdot T^2$$

The final DAC characteristics (**Figure 3**) can then be represented by:

$$Vdac(T, dacin) := dacgain(T) \cdot dacin + dacoffset(T)$$

$$dacin := -0.8, -0.79 .. 0.8$$

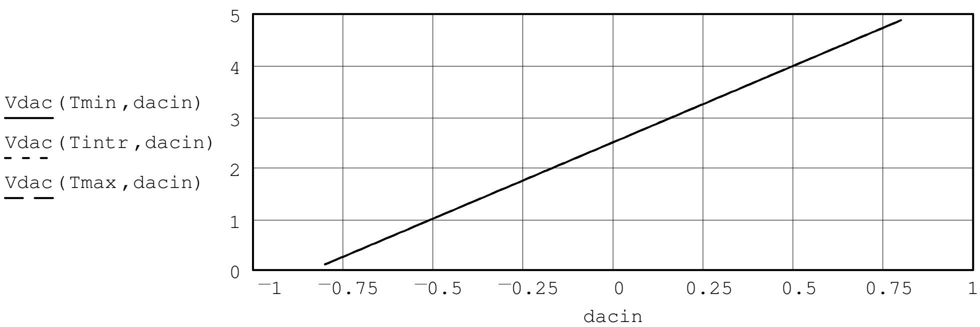

Figure 3. DAC output (V) as a function of normalized input.

For verification, the values below show the input DAC data, and the values obtained from the Vdac function.

$$DacM = \begin{pmatrix} 1.004 \\ 1.0047 \\ 1.0053 \end{pmatrix} \qquad DacP = \begin{pmatrix} 3.9855 \\ 3.9867 \\ 3.9885 \end{pmatrix}$$

$$Vdac(Tmin, DACinM) = 1.004 \qquad Vdac(Tmin, DACinP) = 3.9855$$

$$Vdac(Tintr, DACinM) = 1.0047 \qquad Vdac(Tintr, DACinP) = 3.9867$$

$$Vdac(Tmax, DACinM) = 1.0053 \qquad Vdac(Tmax, DACinP) = 3.9885$$

3. Temperature-Sensor Offset and Nonlinearity Correction

To minimize the temperature-related coefficients, we arrange the temperature characteristics, centering it at zero and then amplifying it.

To center the data points, the temperature data offset is defined as:

$$Toff := \frac{Tdata(Tmin) - Tdata(Tmax)}{2} - Tdata(Tmin)$$

$$Toff = 0.01052856$$

The offset-corrected Tdata (**Figure 4**) is then given by:

$$OCTdata(T) := Tdata(T) + Toff$$

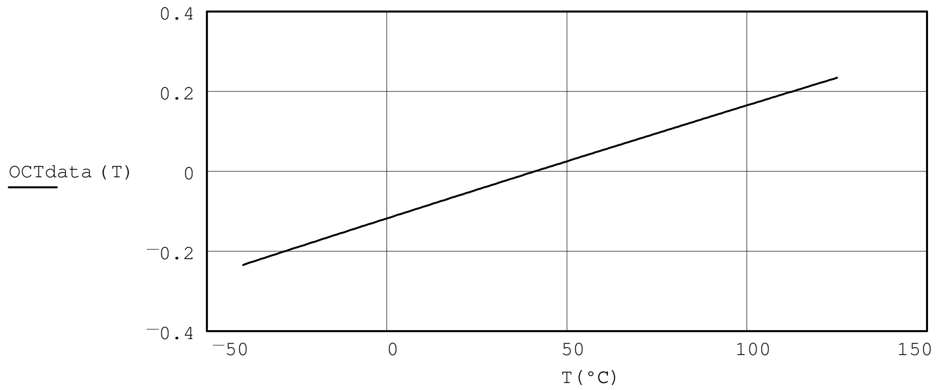

Figure 4. Offset-corrected temperature data.

The next step is to expand this function so that its minimum and maximum values are -0.9 and +0.9 (90% of the useful range). This is done to reduce the values of the temperature-related coefficients that will be calculated in this algorithm. The temperature gain is given by:

$$rtgain := \frac{0.9 - (-0.9)}{OCTdata(Tmax) - OCTdata(Tmin)}$$

$$rtgain = 3.84400417$$

As rtgain is outside the -1 to +1 values, it needs to be scaled down by a power of 2.

```
ntgainshfts :=  | gain ← |rtgain|
                | ntemp ← 1
                | while  gain > 1
                |     | gain ← gain / 2
                |     | ntemp ← ntemp·2
                | log(ntemp) / log(2)
```

$$ntgainshfts = 2$$

The final tgain is then:

$$tgain := \frac{rtgain}{2^{ntgainshfts}}$$

$$tgain = 0.96100104$$

After multiplying tgain by OCTdata, the result needs to be scaled back up, shifting the result to the left (multiplying by powers of 2) by the same factor used in the downscaling process (ntgainshfts). The final amplified, offset-corrected temperature data **(Figure 5)** is then given by:

$$AOCTdata(T) := 2^{ntgainshfts} \cdot tgain \cdot OCTdata(T)$$

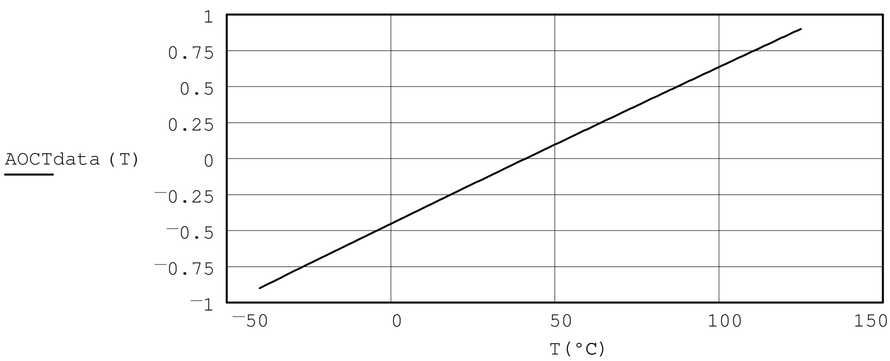

Figure 5. Amplified, offset-corrected temperature data.

where:

$$AOCTdata(Tmin) = -0.9 \qquad AOCTdata(Tmax) = 0.9$$

The next step is the temperature nonlinearity correction. The linear coefficient of AOCTdata, using its endpoints, is calculated as:

$$mt := \frac{AOCTdata(Tmax) - AOCTdata(Tmin)}{Tmax - Tmin}$$

$$mt = 0.01090909$$

And the nonlinear function (**Figure 6**) can be expressed as:

$$Tnonlinearity(T) := AOCTdata(T) - mt \cdot (T - Tmed)$$

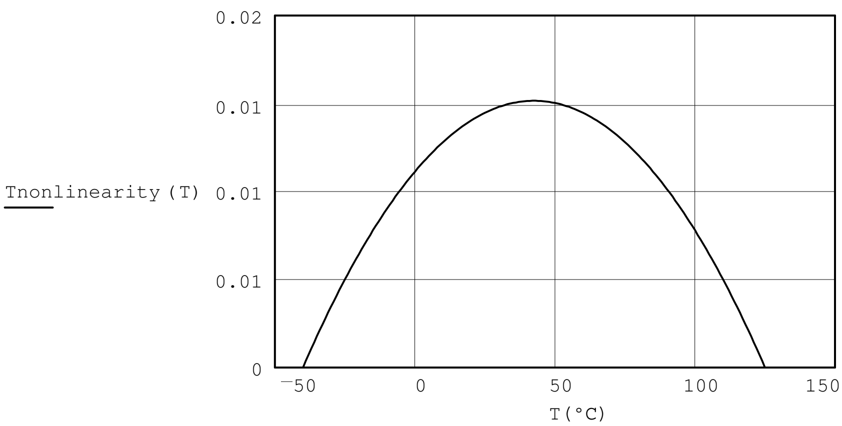

Figure 6. Nonlinearity of temperature data.

Using the AOCTdata(T) as the independent variable, we need to implement a function that represents the opposite of the nonlinearity function. As AOCTdata(T) is not linear, the best way is to use a fitting function to a higher order polynomial function. In this case, a fourth-order polynomial function is chosen to minimize the nonlinearity errors.

$$xdata_{T-Tmin} := AOCTdata(T)$$

$$ydata_{T-Tmin} := -Tnonlinearity(T)$$

The coefficients of the fourth-order polynomial function are given by:

$$tnl := linfit(xdata, ydata, F4)$$

$$tnl = \begin{bmatrix} -0.01515302 \\ -5.67061548 \cdot 10^{-4} \\ 0.0186809 \\ 6.9990991 \cdot 10^{-4} \\ 3.2752246 \cdot 10^{-5} \end{bmatrix}$$

The temperature nonlinearity correction function is given by:

$$Tnl(T) := tnl_0 + tnl_1 \cdot AOCTdata(T) + tnl_2 \cdot AOCTdata(T)^2 \ldots + tnl_3 \cdot AOCTdata(T)^3 + tnl_4 \cdot AOCTdata(T)^4$$

The offset-corrected and nonlinearity-corrected temperature data are now given by:

$$Tempdata(T) := AOCTdata(T) + Tnl(T)$$

From now on, all the temperature-related coefficients will be calculated using Tempdata as the independent variable, as it is normalized and linear (**Figure 7**).

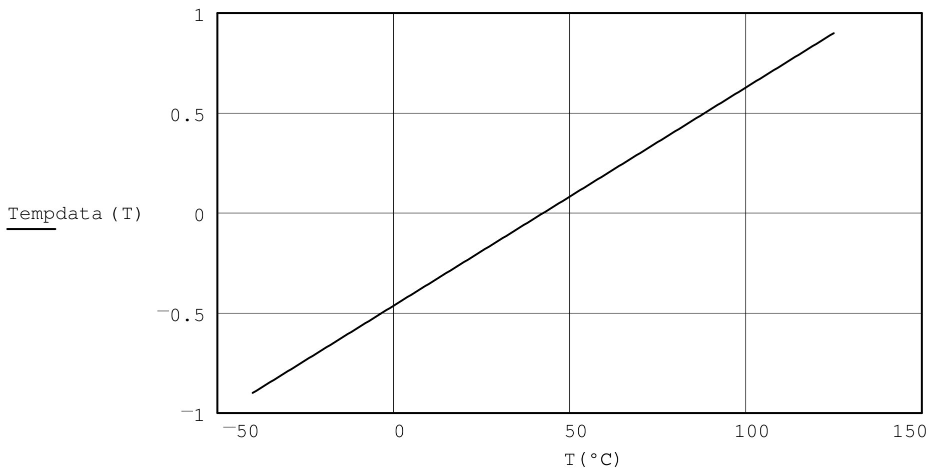

Figure 7. Linearized temperature data.

The ideal temperature data is given by:

$$Ideal(T) := \frac{0.9 - (-0.9)}{Tmax - Tmin} \cdot (T - Tmed)$$

Deviation of Tempdata (T) from an ideal function, Ideal (T), is given in Figure 8.

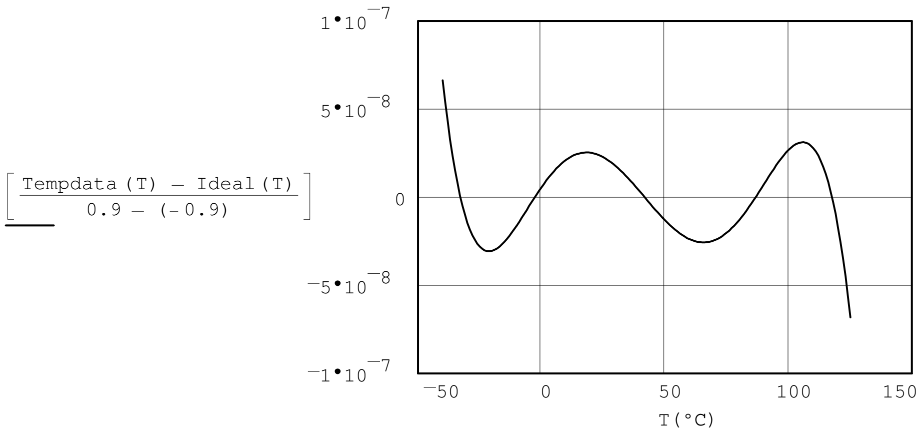

Figure 8. Linearity error of Tempdata(T).

4. Sensor-Signal Data Offset, Gain, and Nonlinearity Correction

The sensor-signal characteristics are also dependent on temperature and the excitation source (pressure). The objective here is to eliminate the temperature dependency and linearize the pressure-response characteristics.

As with the temperature signal, we will maximize the response to 90% of the total useful range, yielding -0.9 for minimum pressure and +0.9 for the maximum pressure.

The raw sensor data is depicted below in **Figure 9**, for four different temperatures:

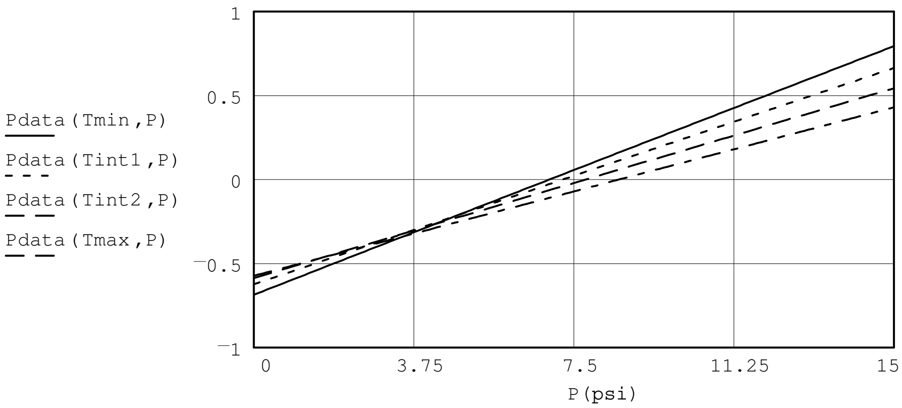

Figure 9. Raw sensor data measured over pressure.

The linear coefficients for these four curves, using the endpoint values, can be given by:

$$mp(T) := \frac{Pdata(T,Pmax) - Pdata(T,Pmin)}{Pmax - Pmin}$$

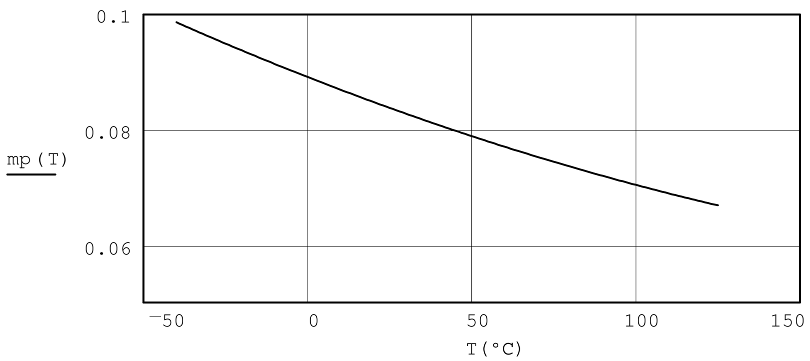

Figure 10. Sensor sensitivity measured over temperature.

The pressure-nonlinearity and offset-correction function (**Figure 11**) can be expressed by:

$$pnl(T,P) := Pdata(T,P) - mp(T)\cdot(P - Pmed)$$

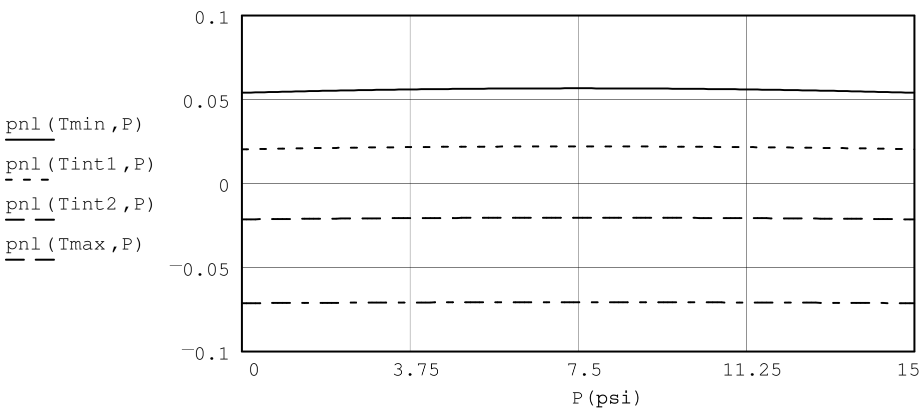

Figure 11. Sensor offset and nonlinearity measured over pressure.

The objective now is to model the reciprocal of the pnl(T,P) function for four different temperatures, and then model the variation of the coefficients over temperature.

$$j := 0 .. 150$$

$$PP_j := Pmin + \frac{Pmax - Pmin}{150}\cdot j$$

For the minimum temperature, the coefficients that approximate -pnl(Tmin,P) over a third-order function of Pdata(Tmin,P) can be found by:

$$axdata_j := Pdata\left(Tmin, PP_j\right)$$

$$aydata_j := -pnl\left(Tmin, PP_j\right)$$

$$acc := linfit(axdata, aydata, F3)$$

$$acc = \begin{pmatrix} -0.05665568 \\ -5.46133588 \cdot 10^{-4} \\ 0.0048145 \\ 4.65054473 \cdot 10^{-5} \end{pmatrix}$$

For the first intermediate temperature, the coefficients that approximate -pnl(Tint1,P) over a third-order function of Pdata(Tint1,P) can be found by:

$$bxdata_j := Pdata\left(Tint1, PP_j\right)$$

$$bydata_j := -pnl\left(Tint1, PP_j\right)$$

$$bcc := linfit(bxdata, bydata, F3)$$

$$bcc = \begin{pmatrix} -0.02208378 \\ -1.76379526 \cdot 10^{-4} \\ 0.00399201 \\ 3.19040552 \cdot 10^{-5} \end{pmatrix}$$

For the second intermediate temperature, the coefficients that approximate -pnl(Tint2,P) over a third-order function of Pdata(Tint2,P) can be found by:

$$cxdata_j := Pdata\left(Tint2, PP_j\right)$$

$$cydata_j := -pnl\left(Tint2, PP_j\right)$$

$$ccc := linfit(cxdata, cydata, F3)$$

$$ccc = \begin{pmatrix} 0.02035532 \\ 1.24568012 \cdot 10^{-4} \\ 0.0030606 \\ 1.87200228 \cdot 10^{-5} \end{pmatrix}$$

For the maximum temperature, the coefficients that approximate -pnl(Tmax,P) over a third-order function of Pdata(Tmax,P) can be found by:

$$dxdata_j := Pdata\left(Tmax, PP_j\right)$$

$$dydata_j := -pnl\left(Tmax, PP_j\right)$$

$$dcc := linfit(dxdata, dydata, F3)$$

$$dcc = \begin{pmatrix} 0.07066001 \\ 3.34484394 \cdot 10^{-4} \\ 0.00236844 \\ 1.11963953 \cdot 10^{-5} \end{pmatrix}$$

The matrix of correction coefficients for the calculated temperatures is given below:

$$ccoef := \begin{pmatrix} acc_0 & acc_1 & acc_2 & acc_3 \\ bcc_0 & bcc_1 & bcc_2 & bcc_3 \\ ccc_0 & ccc_1 & ccc_2 & ccc_3 \\ dcc_0 & dcc_1 & dcc_2 & dcc_3 \end{pmatrix}$$

$$ccoef = \begin{pmatrix} -0.05665568 & -5.46133588\cdot10^{-4} & 0.0048145 & 4.65054473\cdot10^{-5} \\ -0.02208378 & -1.76379526\cdot10^{-4} & 0.00399201 & 3.19040552\cdot10^{-5} \\ 0.02035532 & 1.24568012\cdot10^{-4} & 0.0030606 & 1.87200228\cdot10^{-5} \\ 0.07066001 & 3.34484394\cdot10^{-4} & 0.00236844 & 1.11963953\cdot10^{-5} \end{pmatrix}$$

The variation of those coefficients over the temperature data can be modeled by third-order equations, solving a linear system:

$$Tinv2 := \begin{pmatrix} 1 & Tempdata(Tmin) & Tempdata(Tmin)^2 & Tempdata(Tmin)^3 \\ 1 & Tempdata(Tint1) & Tempdata(Tint1)^2 & Tempdata(Tint1)^3 \\ 1 & Tempdata(Tint2) & Tempdata(Tint2)^2 & Tempdata(Tint2)^3 \\ 1 & Tempdata(Tmax) & Tempdata(Tmax)^2 & Tempdata(Tmax)^3 \end{pmatrix}^{-1}$$

$$tcc := Tinv2\cdot ccoef$$

The correction coefficients are:

$$tcc = \begin{pmatrix} -0.00184753 & -1.59159026\cdot10^{-5} & 0.00351816 & 2.48696787\cdot10^{-5} \\ 0.07073196 & 5.03122682\cdot10^{-4} & -0.00157653 & -2.22680466\cdot10^{-5} \\ 0.01092556 & -1.10998417\cdot10^{-4} & 9.05029565\cdot10^{-5} & 4.91511553\cdot10^{-6} \\ -1.25221931\cdot10^{-6} & -1.71486526\cdot10^{-5} & 2.68656352\cdot10^{-4} & 3.27395645\cdot10^{-6} \end{pmatrix}$$

The zero-, first-, second-, and third-order coefficient functions are given by:

$$NL0(T) := tcc_{0,0} + tcc_{1,0}\cdot Tempdata(T) + tcc_{2,0}\cdot Tempdata(T)^2 + tcc_{3,0}\cdot Tempdata(T)^3$$

$$NL1(T) := tcc_{0,1} + tcc_{1,1}\cdot Tempdata(T) + tcc_{2,1}\cdot Tempdata(T)^2 + tcc_{3,1}\cdot Tempdata(T)^3$$

$$NL2(T) := tcc_{0,2} + tcc_{1,2}\cdot Tempdata(T) + tcc_{2,2}\cdot Tempdata(T)^2 + tcc_{3,2}\cdot Tempdata(T)^3$$

$$NL3(T) := tcc_{0,3} + tcc_{1,3}\cdot Tempdata(T) + tcc_{2,3}\cdot Tempdata(T)^2 + tcc_{3,3}\cdot Tempdata(T)^3$$

The offset- and nonlinearity-correction function is then given by:

$$OLC(T,P) := NL0(T) + NL1(T)\cdot Pdata(T,P) + NL2(T)\cdot Pdata(T,P)^2 + NL3(T)\cdot Pdata(T,P)^3$$

The final offset- and nonlinearity-corrected sensor data is **(Figure 12)** now given by:

$$OLCPdata(T,P) := Pdata(T,P) + OLC(T,P)$$

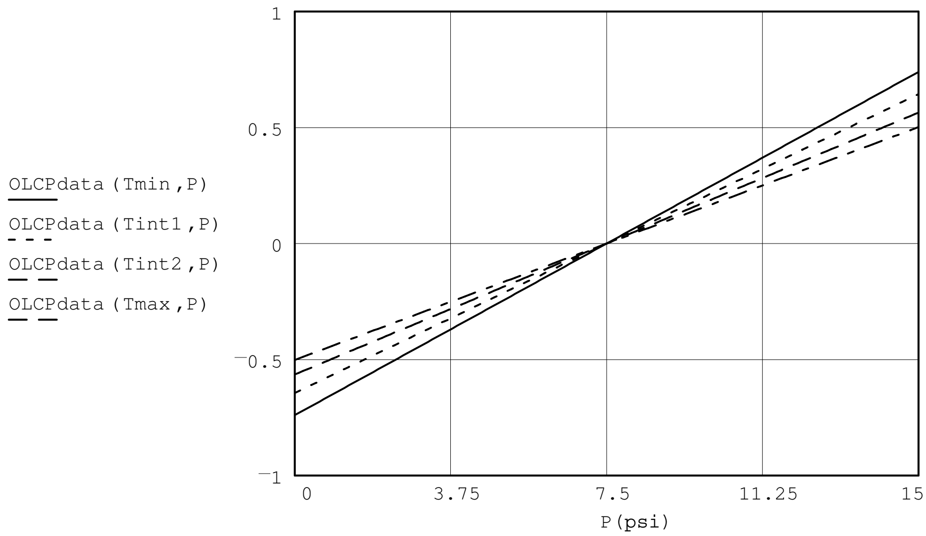

Figure 12. Offset and nonlinearity-corrected sensor data.

The next step is to remove the temperature dependency of the sensor sensitivity.

The span function over temperature is given by:

$$span(T) := OLCPdata(T,Pmax) - OLCPdata(T,Pmin)$$

The sensitivity correction coefficients that approximate 1/span(T) over a fourth order function of Tempdata(T) can be found by:

$$xdata_{T-Tmin} := Tempdata(T)$$

$$ydata_{T-Tmin} := \frac{1}{span(T)}$$

$$sc := linfit(xdata, ydata, F4)$$

$$sc = \begin{pmatrix} 0.83011994 \\ 0.18253006 \\ 0.0088296 \\ -0.00503055 \\ -0.00135329 \end{pmatrix}$$

The sensitivity correction function (**Figure 13**) can then be described by:

$$SensCorrection(T) := sc_0 + sc_1 \cdot Tempdata(T) + sc_2 \cdot Tempdata(T)^2 \ldots + sc_3 \cdot Tempdata(T)^3 + sc_4 \cdot Tempdata(T)^4$$

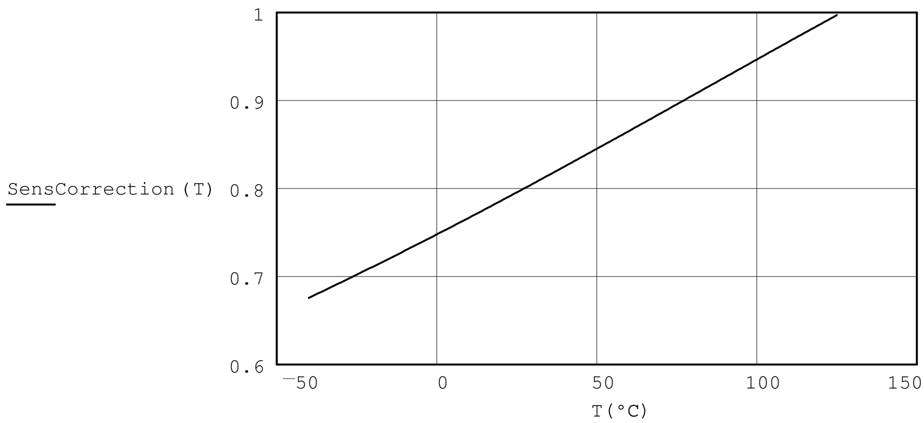

Figure 13. Sensitivity correction function measured over temperature. (ºC)

The corrected sensor data is then given by:

$$CPdata(T,P) := SensCorrection(T) \cdot OLCPdata(T,P)$$

The final data will be normalized to -0.9 for the minimum pressure and +0.9 for the maximum pressure. The normalization factor is given by:

$$nfactor := \frac{0.9 - (-0.9)}{CPdata(Tmed,Pmax) - CPdata(Tmed,Pmin)}$$

$$nfactor = 1.79999926$$

The final coefficients for the sensitivity-correction function are then given by:

$$nsc := nfactor \cdot sc$$

$$nsc = \begin{pmatrix} 1.49421527 \\ 0.32855396 \\ 0.01589327 \\ -0.00905498 \\ -0.00243592 \end{pmatrix}$$

Where the normalized sensitivity-correction function is given by:

$$nSensC(T) := nsc_0 + nsc_1 \cdot Tempdata(T) + nsc_2 \cdot Tempdata(T)^2 \ldots + nsc_3 \cdot Tempdata(T)^3 + nsc_4 \cdot Tempdata(T)^4$$

The value of nSensC(T) must be between -1 and +1 for the whole temperature range. To ensure that this is true, we have to find the power-of-two divisor that scales back nSensC(T).

$$nSensVector_{T-Tmin} := |nSensC(T)|$$

$$max(nSensVector) = 1.79458807$$

```
nsgainshfts :=  | gain ← max(nSensVector)
                | ntemp ← 1
                | while  gain > 1
                |     | gain ← gain/2
                |     | ntemp ← ntemp·2
                | log(ntemp)/log(2)
```

$$nsgainshfts = 1$$

The final set of sensitivity-correction coefficients is then given by:

$$fnsc := \frac{nsc}{2^{nsgainshfts}}$$

$$fnsc = \begin{pmatrix} 0.74710764 \\ 0.16427698 \\ 0.00794664 \\ -0.00452749 \\ -0.00121796 \end{pmatrix}$$

The final sensitivity-correction function is then given by:

$$fnSensC(T) := fnsc_0 + fnsc_1 \cdot Tempdata(T) + fnsc_2 \cdot Tempdata(T)^2 \ldots + fnsc_3 \cdot Tempdata(T)^3 + fnsc_4 \cdot Tempdata(T)^4$$

The final normalized-corrected sensor data (**Figure 14)** is then given by:

$$nCPdata(T,P) := 2^{nsgainshfts} \cdot fnSensC(T) \cdot OLCPdata(T,P)$$

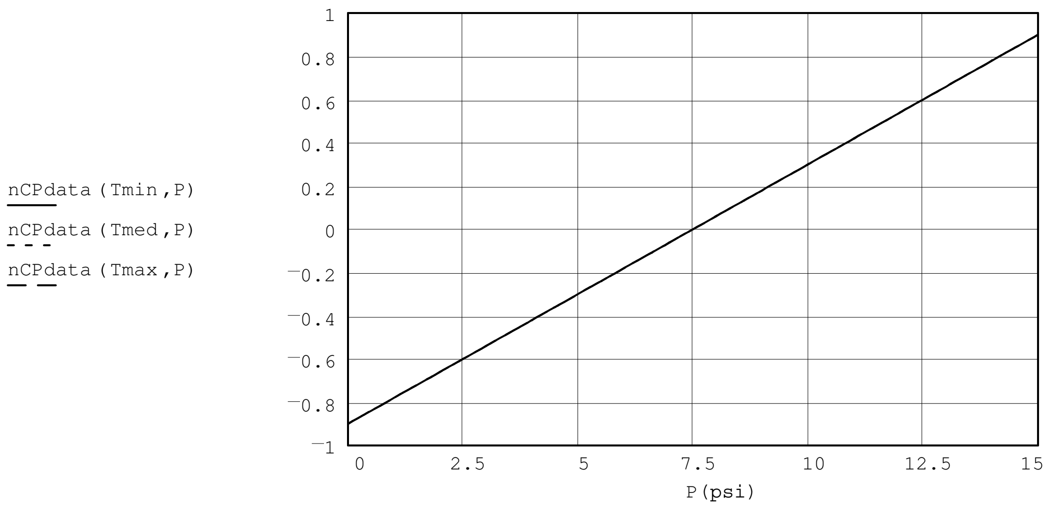

Figure 14. Normalized corrected sensor data measured against pressure.

At this point, the sensor data is normalized to -0.9 to +0.9 for the minimum and maximum, respectively, sensor excitation. All the nonlinearities have been corrected and the temperature dependency is removed. It is a very linear signal with respect to the excitation (pressure).

5. DAC Correction

When an analog output is required, this step corrects the nonlinearities and temperature dependency associated with the MAX1464 DACs. The minimum and maximum output voltages associated with the minimum and maximum sensor excitation were already defined. They are given by:

$$Vmin = 0.5$$

$$Vmax = 4.5$$

The offset can then be defined as:

$$Offset := \frac{Vmax - Vmin}{2} + Vmin$$

$$Offset = 2.5$$

But the target offset value has to compensate for the DAC offset variation over temperature, which is given by:

$$targetOffset(T) := \frac{Offset - dacoffset(T)}{dacgain(T)}$$

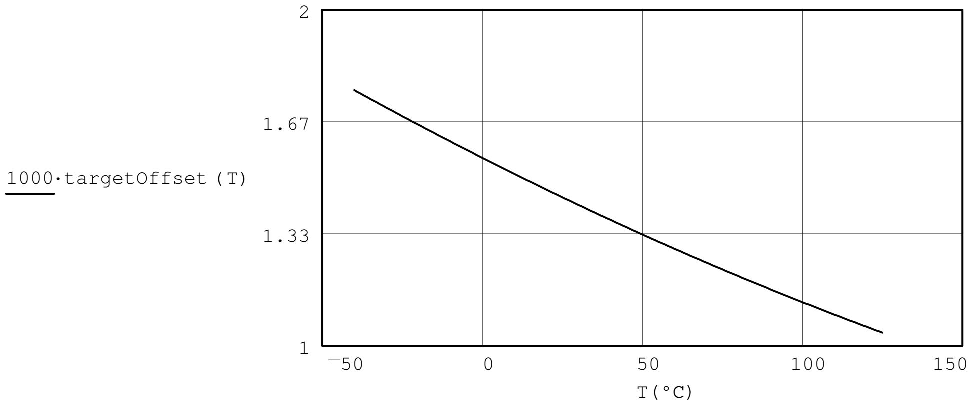

Figure 15. Target DAC correction over temperature.

The DAC offset-correction coefficients that approximate targetOffset(T) over a second-order function of Tempdata(T) can be found by:

$$xdata_{T-Tmin} := Tempdata(T)$$

$$ydata_{T-Tmin} := targetOffset(T)$$

$$doc := linfit(xdata, ydata, F2)$$

$$doc = \begin{pmatrix} 0.00136381 \\ -4.00952681 \cdot 10^{-4} \\ 4.46867159 \cdot 10^{-5} \end{pmatrix}$$

The DAC offset-correction function can then be described by:

$$dacOffsetCorrection(T) := doc_0 + doc_1 \cdot Tempdata(T) + doc_2 \cdot Tempdata(T)^2$$

The DAC gain must also be corrected for temperature variations. The function that corrects the DAC gain variations over temperature and adjusts the signals for the output span is given by (**Figure 16**):

$$targetGain(T) := \frac{Vmax - Vmin}{0.9 - (-0.9)} \cdot \frac{1}{dacgain(T)}$$

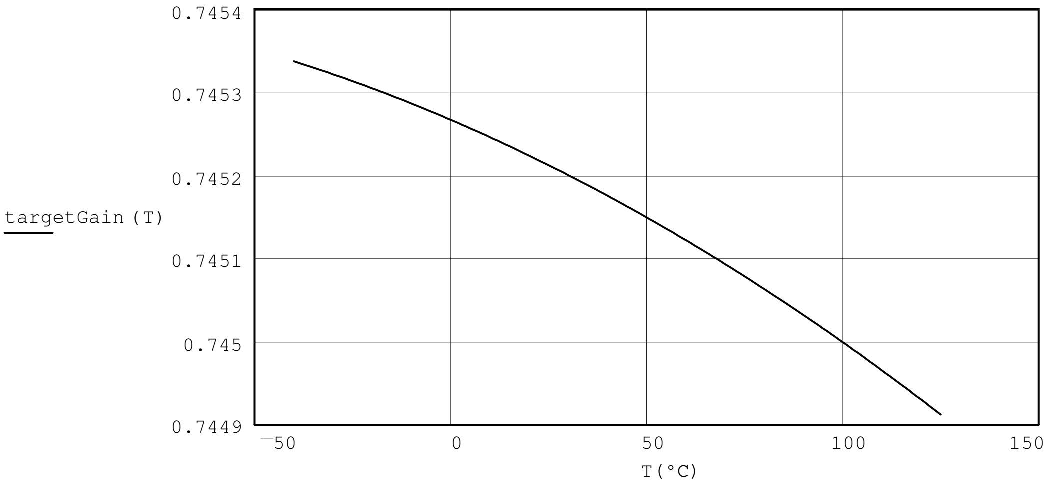

Figure 16. DAC gain correction measured over temperature.

The DAC gain-correction coefficients that approximate targetGain(T) over a second-order function of Tempdata(T) can be found by:

$$xdata_{T-Tmin} := \mathrm{Tempdata}(T)$$

$$ydata_{T-Tmin} := \mathrm{targetGain}(T)$$

$$dgc := \mathrm{linfit}(xdata, ydata, F2)$$

$$dgc = \begin{pmatrix} 0.74516895 \\ -2.35975563 \cdot 10^{-4} \\ -5.47382243 \cdot 10^{-5} \end{pmatrix}$$

And the DAC gain correction function can then be described by:

$$\mathrm{dacGainCorrection}(T) := dgc_0 + dgc_1 \cdot \mathrm{Tempdata}(T) + dgc_2 \cdot \mathrm{Tempdata}(T)^2$$

The final DAC input, as a function of temperature and pressure, is then given by:

$$\mathrm{dacin}(T,P) := \mathrm{dacGainCorrection}(T) \cdot \mathrm{nCPdata}(T,P) + \mathrm{dacOffsetCorrection}(T)$$

The final DAC output over the sensor excitation (pressure) is shown below in **Figure 17**, for various temperatures.

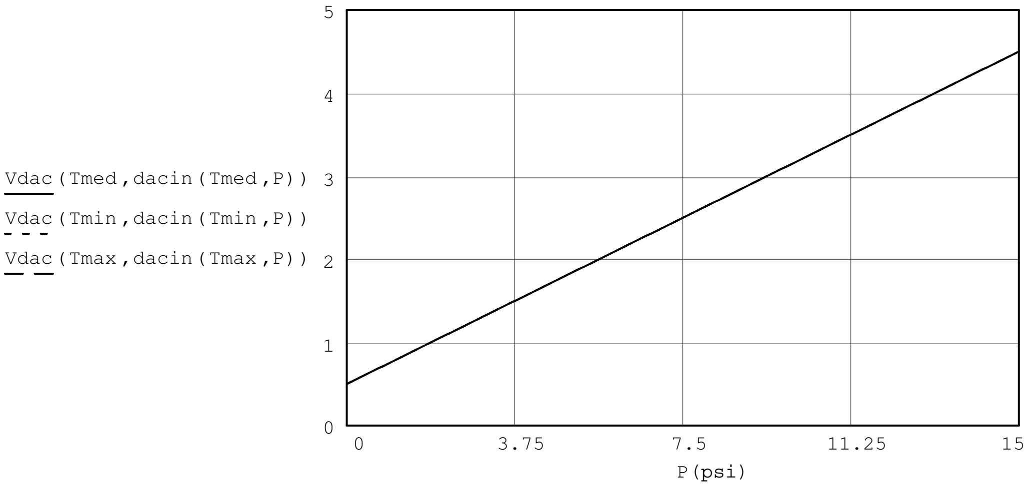

Figure 17. DAC output (V) over pressure.

6. Compensation Coefficients and Equations

This section summarizes the compensation coefficients and equations that need to be implemented in the MAX1464. Note that the MAX1464 does not calculate the coefficients; they need to be calculated using the algorithm described in this document.

The temperature-sensor data is shown as Tdata(T), and is the result of the ADC conversion of the internal MAX1464 temperature sensor. The sensor-signal data is shown as Pdata(T), and is the result of the ADC conversion of the sensor signal.

The program needs both the temperature-sensor data, and the sensor-signal data. As the temperature rate of change is much slower than the sensor-signal data, the user can perform an ADC temperature conversion on a much slower rate than the sensor signal, typically once every few hundred sensor-signal conversions.

The following functions are defined to convert the calculated coefficients to a two's complement hexadecimal representation. Note that the digitized coefficients may differ from the calculated values due to quantization on a 16-bit level.

$$\mathrm{sign}(x) := \mathrm{if}(x < 0, 0, 1)$$

$$\mathrm{ip}(x) := \begin{cases} \mathrm{if}(|\mathrm{ceil}(|x|) - |x|| > |\mathrm{floor}(|x|) - |x||, \mathrm{floor}(x), \mathrm{ceil}(x)) & \text{if } x > 0 \\ \mathrm{if}(|\mathrm{ceil}(|x|) - |x|| > |\mathrm{floor}(|x|) - |x||, \mathrm{ceil}(x), \mathrm{floor}(x)) & \text{if } x < 0 \end{cases}$$

$$\mathrm{d2hi}(d) := \mathrm{if}\left[|d| < 1, \left(\mathrm{ip}\left(2^{15} \cdot d\right) - \mathrm{sign}(d) \cdot 2^{16}\right) + 2^{16}, \mathrm{ip}(d) - \mathrm{sign}(d) \cdot 2^{16} + 2^{16}\right]$$

$$\mathrm{d2h}(d) := \mathrm{if}(\mathrm{d2hi}(d) = 65536, 0, \mathrm{d2hi}(d))$$

6.1. Compensation Coefficients

Toff = 0.01052856 d2h(Toff) = 159h

tgain = 0.96100104 d2h(tgain) = 7b02h

ntgainshfts = 2 d2h(ntgainshfts) = 2h

$$tnl = \begin{bmatrix} -0.01515302 \\ -5.67061548\cdot 10^{-4} \\ 0.0186809 \\ 6.9990991\cdot 10^{-4} \\ 3.2752246\cdot 10^{-5} \end{bmatrix}$$

d2h(tnl_0) = fe0fh
d2h(tnl_1) = ffedh
d2h(tnl_2) = 264h
d2h(tnl_3) = 17h
d2h(tnl_4) = 1h

$$tcc = \begin{pmatrix} -0.00184753 & -1.59159026\cdot 10^{-5} & 0.00351816 & 2.48696787\cdot 10^{-5} \\ 0.07073196 & 5.03122682\cdot 10^{-4} & -0.00157653 & -2.22680466\cdot 10^{-5} \\ 0.01092556 & -1.10998417\cdot 10^{-4} & 9.05029565\cdot 10^{-5} & 4.91511553\cdot 10^{-6} \\ -1.25221931\cdot 10^{-6} & -1.71486526\cdot 10^{-5} & 2.68656352\cdot 10^{-4} & 3.27395645\cdot 10^{-6} \end{pmatrix}$$

d2h$(tcc_{0,0})$ = ffc3h d2h$(tcc_{0,1})$ = ffffh d2h$(tcc_{0,2})$ = 73h d2h$(tcc_{0,3})$ = 1h
d2h$(tcc_{1,0})$ = 90eh d2h$(tcc_{1,1})$ = 10h d2h$(tcc_{1,2})$ = ffcch d2h$(tcc_{1,3})$ = ffffh
d2h$(tcc_{2,0})$ = 166h d2h$(tcc_{2,1})$ = fffch d2h$(tcc_{2,2})$ = 3h d2h$(tcc_{2,3})$ = 0
d2h$(tcc_{3,0})$ = 0 d2h$(tcc_{3,1})$ = ffffh d2h$(tcc_{3,2})$ = 9h d2h$(tcc_{3,3})$ = 0

$$fnsc = \begin{pmatrix} 0.74710764 \\ 0.16427698 \\ 0.00794664 \\ -0.00452749 \\ -0.00121796 \end{pmatrix}$$

d2h$(fnsc_0)$ = 5fa1h
d2h$(fnsc_1)$ = 1507h
d2h$(fnsc_2)$ = 104h
d2h$(fnsc_3)$ = ff6ch
d2h$(fnsc_4)$ = ffd8h

$$doc = \begin{pmatrix} 0.00136381 \\ -4.00952681\cdot 10^{-4} \\ 4.46867159\cdot 10^{-5} \end{pmatrix}$$

d2h(doc_0) = 2dh
d2h(doc_1) = fff3h
d2h(doc_2) = 1h

$$dgc = \begin{pmatrix} 0.74516895 \\ -2.35975563\cdot 10^{-4} \\ -5.47382243\cdot 10^{-5} \end{pmatrix}$$

d2h(dgc_0) = 5f62h
d2h(dgc_1) = fff8h
d2h(dgc_2) = fffeh

nsgainshfts = 1

6.2. Temperature-Loop Compensation Equations

The following set of equations corrects the MAX1464 temperature-sensor data.

$$OCTdata(T) := Tdata(T) + Toff$$

$$AOCTdata(T) := 2^{ntgainshfts} \cdot tgain \cdot OCTdata(T)$$

$$Tnl(T) := tnl_0 + tnl_1 \cdot AOCTdata(T) + tnl_2 \cdot AOCTdata(T)^2 \ldots + tnl_3 \cdot AOCTdata(T)^3 + tnl_4 \cdot AOCTdata(T)^4$$

$$Tempdata(T) := AOCTdata(T) + Tnl(T)$$

The next set of equations corrects the sensor-signal data.

$$NL0(T) := tcc_{0,0} + tcc_{1,0} \cdot Tempdata(T) + tcc_{2,0} \cdot Tempdata(T)^2 + tcc_{3,0} \cdot Tempdata(T)^3$$

$$NL1(T) := tcc_{0,1} + tcc_{1,1} \cdot Tempdata(T) + tcc_{2,1} \cdot Tempdata(T)^2 + tcc_{3,1} \cdot Tempdata(T)^3$$

$$NL2(T) := tcc_{0,2} + tcc_{1,2} \cdot Tempdata(T) + tcc_{2,2} \cdot Tempdata(T)^2 + tcc_{3,2} \cdot Tempdata(T)^3$$

$$NL3(T) := tcc_{0,3} + tcc_{1,3} \cdot Tempdata(T) + tcc_{2,3} \cdot Tempdata(T)^2 + tcc_{3,3} \cdot Tempdata(T)^3$$

$$fnSensC(T) := fnsc_0 + fnsc_1 \cdot Tempdata(T) + fnsc_2 \cdot Tempdata(T)^2 \ldots + fnsc_3 \cdot Tempdata(T)^3 + fnsc_4 \cdot Tempdata(T)^4$$

The next set of coefficients and equations provide correction on the MAX1464 DAC data.

$$dacOffsetCorrection(T) := doc_0 + doc_1 \cdot Tempdata(T) + doc_2 \cdot Tempdata(T)^2$$

$$dacGainCorrection(T) := dgc_0 + dgc_1 \cdot Tempdata(T) + dgc_2 \cdot Tempdata(T)^2$$

6.3. Sensor Signal Loop Compensation Equations

The following set of equations corrects the MAX1464's pressure-sensor data.

$$OLC(T,P) := NL0(T) + NL1(T) \cdot Pdata(T,P) + NL2(T) \cdot Pdata(T,P)^2 + NL3(T) \cdot Pdata(T,P)^3$$

$$OLCPdata(T,P) := Pdata(T,P) + OLC(T,P)$$

$$nCPdata(T,P) := 2^{nsgainshfts} \cdot fnSensC(T) \cdot OLCPdata(T,P)$$

$$dacin(T,P) := dacGainCorrection(T) \cdot nCPdata(T,P) + dacOffsetCorrection(T)$$

At this point, just write the final result, dacin(T,P), to the DAC input to obtain the compensated output. Return to the beginning of the sensor-signal loop. The user may implement a counter to track the number of sensor-signal conversions, and do a temperature conversion every so often.

7. References

7.1. To view the MAX1464 data sheet, visit: www.maxim-ic.com/MAX1464
7.2. See application note, "Understanding the MAX1464 On-Chip Temperature Sensor"
7.3. See application Note, "An Embedded Compensation Program for the MAX1463 High-Performance Signal Conditioner"
7.4. For more information on Maxim's thermal management, sensors, and signal conditioners, visit: www.maxim-ic.com/Sensors.cfm