

TMS320280x and TMS3202801x ADC Calibration

Matthew Pate

AEC C2000

ABSTRACT

This application report describes a method for improving the absolute accuracy of the 12-bit ADC found on the TMS320280x and TMS3202801x devices. Inherent gain and offset errors affect the absolute accuracy of the ADC. The methods described in this report can improve the absolute accuracy of the ADC to levels better than 0.5%.

This application report has an option to download an example program that executes from RAM on the F2808 EzDSP. See the following link for the download:
<http://www.ti.com/lit/zip/SPRAAD8>.

Note: The data in this application report is based on observations of ADC behavior on TMS320280x Rev-B TMS devices.

Contents

1	Introduction	2
2	Gain and Offset Error Definition.....	2
3	Gain and Offset Error Impact.....	4
4	Calibration.....	6
5	Hardware Connectivity.....	8
6	ADC Sampling Techniques.....	11
7	Example Software Calibration Driver	15
8	Gain and Offset Temperature Drift	17
9	Useful Tips.....	18

List of Figures

1	Gain and Offset Definition.....	3
2	Simplified Circuit for Ideal ADC Case	5
3	Modified Circuit to Scale Input to Safe Range of ADC	5
4	ADC Actual Gain.....	6
5	Channels for Calibration, 14 User Channels	8
6	2 Channels for Calibration, 16 User Channels	10
7	Timing Details of 16 Channel Conversion	11
8	Channel Selection Controls	12
9	Timing Details of 8 Pairs of Channels	13
10	Channel Selection.....	14
11	Gain Drift Over Temperature	17
12	Offset Drift Over Temperature	18

List of Tables

1	Input Voltage Range.....	4
2	Conversion Time of All 16 Channels	11
3	Converting 16 Channels in Simultaneous Mode.....	13

1 Introduction

This application report includes the following sections:

- Gain and offset error definition
- Gain and offset error impact
- Calibration
- Hardware connectivity
- ADC sampling techniques
- Example software calibration driver
- Gain and offset temperature drift
- Useful tips

2 Gain and Offset Error Definition

An ideal 12-bit ADC with no gain and offset error is described by [Equation 1](#):

$$y = x \xi m_i$$

x = input count = input voltage * 4095/3.0 V
 y = output count
 m_i = ideal gain = 1.0000

(1)

The F280x ADC exhibits gain and offset error as defined in [Equation 2](#):

$$y = (x \xi m_a) + b$$

m_a = actual gain
 b = actual offset (relative to 0 input)

(2)

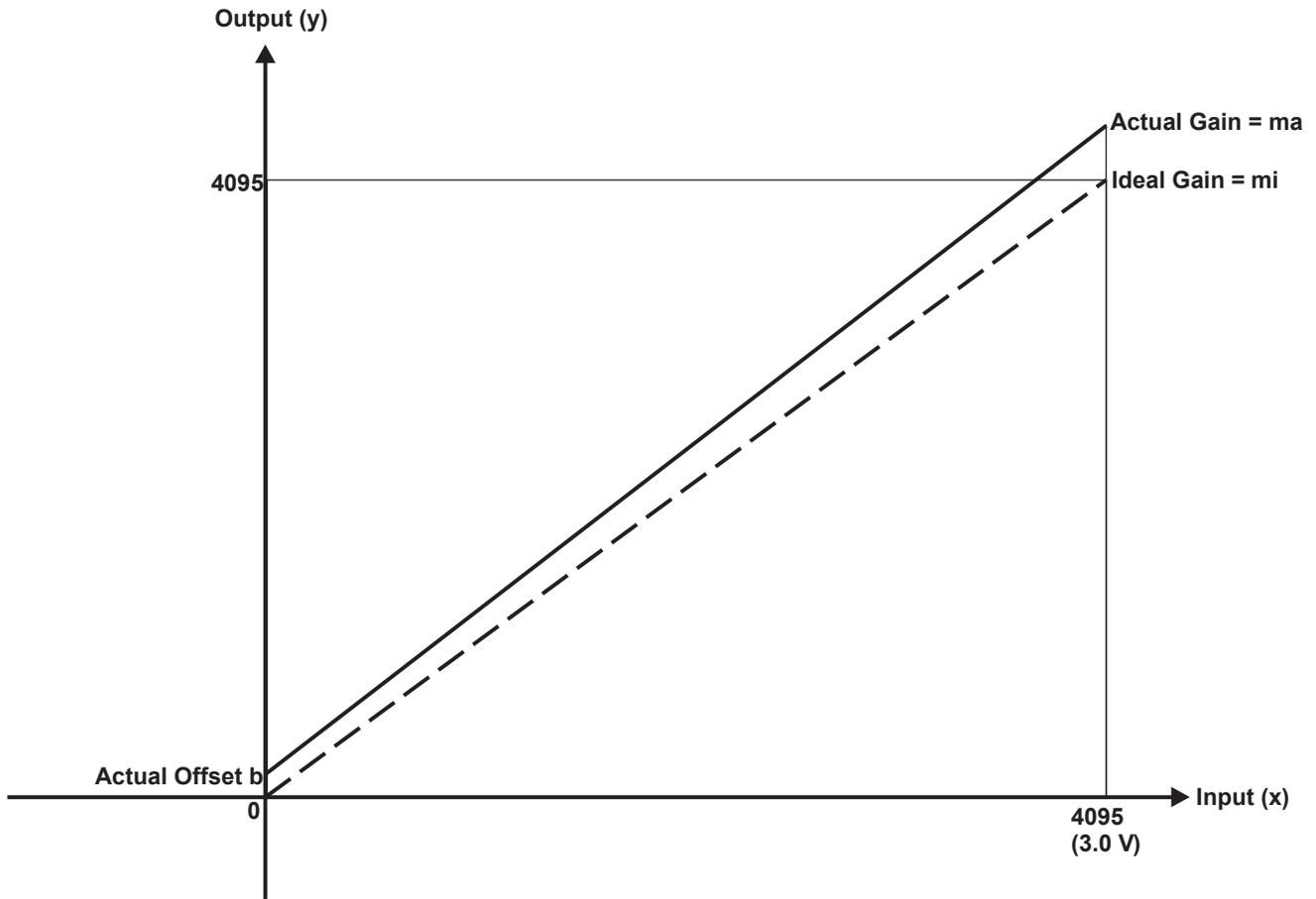


Figure 1. Gain and Offset Definition

Gain and offset errors measured on the F280x ADC are:

Error	Measure	Range
Gain Error (m_a)	$< \pm 1.5\%$ max	$0.985 < m_a < 1.015$
Offset Error (b)	$< \pm 1.5\%$ max	$-60 < b < 60$

Note: Most F280x devices exhibit gain errors of $< \pm 1\%$ and offset errors of $< \pm 1\%$. However, some devices exhibit errors above these values, reaching the maximum values outlined above and in the datasheet.

3 Gain and Offset Error Impact

Gain and offset errors contribute to errors in the control system; however, you can compensate for them in the design.

Linear Input Range: The available input voltage range is affected by the gain and offset errors and the effective resolution is reduced. [Table 1](#) summarizes the worst case scenarios:

Table 1. Input Voltage Range

	Linear Input Range (Volts)	Linear Output Range (counts)	Input Swing (Volts)	Effective Number of Bits	mV/ Count Resolution
$y = x \times 1.00$	0.0000 to 3.0000	0 to 4095	1.5000 ± 1.5000	12.000	0.7326
$y = x \times 1.015+10$	0.0000 to 2.9487	10 to 4095	1.4744 ± 1.4744	11.996	0.7218
$y = x \times 0.985+10$	0.0000 to 3.0000	10 to 4044	1.5000 ± 1.5000	11.978	0.7437
$y = x \times 1.015+15$	0.0000 to 2.9450	15 to 4095	1.4725 ± 1.4725	11.994	0.7218
$y = x \times 0.985+15$	0.0000 to 3.0000	15 to 4049	1.5000 ± 1.5000	11.978	0.7353
$y = x \times 1.015+5$	0.0000 to 2.9546	5 to 4095	1.4773 ± 1.4773	11.998	0.7224
$y = x \times 0.985+5$	0.0000 to 3.0000	5 to 4039	1.5000 ± 1.5000	11.978	0.7437
Safe Range	0.0000 to 2.9450	15 to 4039	1.4725 ± 1.4725	11.978	0.7319

Worst case offsets in the above table are not indicative of the values published in the F280x datasheet of ± 60 least significant bits (LSBs). While the offset inherent to the ADC module can equal ± 60 LSBs, a new feature in the F280x ADC can cancel out the offset.

The offset cancellation is handled via a 9 bit register, ADCOFFTRIM, capable of changing the offset of a measurement by $-256/+255$ (2's complement format, see the *TMS320x280x 2801x, 2804x Analog to Digital Converter (ADC) Module Reference Guide (SPRU716)*). The value in this register changes the analog characteristics of the ADC, allowing full offset compensation while maintaining the native range of the ADC. This differs from a pure digital offset correction handled post conversion, which would take extra CPU cycles and not improve the range of the ADC.

For example, if the offset produced by the ADC was +40 LSB (assuming unity gain), then the max voltage that could be input before saturation would be $4095-40 = 4055$ LSB or $4055 \text{ LSB} \times 0.0007326 \text{ V/LSB} = 2.97 \text{ V}$, for example, all voltages above 2.97 V would give code 4095. If the offset cancellation register is loaded with -40, it effectively increases the range of the ADC such that voltages up to 3.00 V can be input with no early saturation.

Based on the above, it may be unclear why offset values of 5/10/15 are used to calculate the safe range of the ADC. While offset can be effectively eliminated, there are board and environment factors that may alter the offset by a few LSBs. If the offset becomes negative, the ability to differentiate between 0.00 V and the next few steps becomes impossible. That ability is important to many applications, so you must set the offset cancellation register to produce an artificial offset of +10 LSB. This protects from the loss of 0.00 V detection while having little impact on top end performance. If complete offset cancellation is required, you can reinitialize the offset correction register.

The last row in the table shows the safe parameters under which all devices can be guaranteed to operate. The effective number of bits of the ADC is only slightly reduced (11.978 bits). The reduction in voltage range has a slight impact on external noise sensitivity. The mV/count is reduced from 0.7326 to 0.7319, which is about a 0.1% reduction (or a 0.1% increase in sensitivity over the ideal case). With the use of offset cancellation, 0 V is assured to be in the safe range of input voltage.

Bipolar Offset Error: In many applications, the input sensor is a bipolar input that must be converted to a unipolar signal before being fed to the ADC. Figure 2 shows a typical simplified circuit used for this purpose (ideal ADC case).

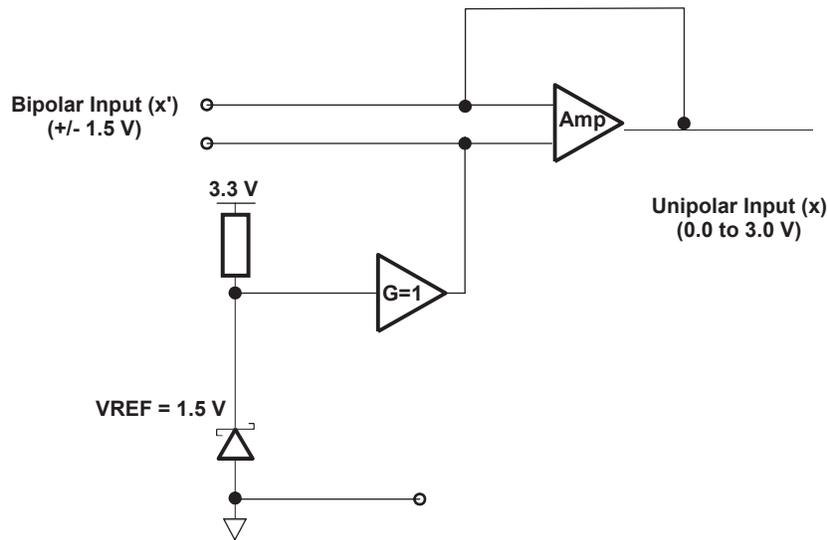


Figure 2. Simplified Circuit for Ideal ADC Case

To allow for gain and offset errors and the impact on the input range, you can modify the circuit as shown in Figure 3 to account for all possible device characteristics:

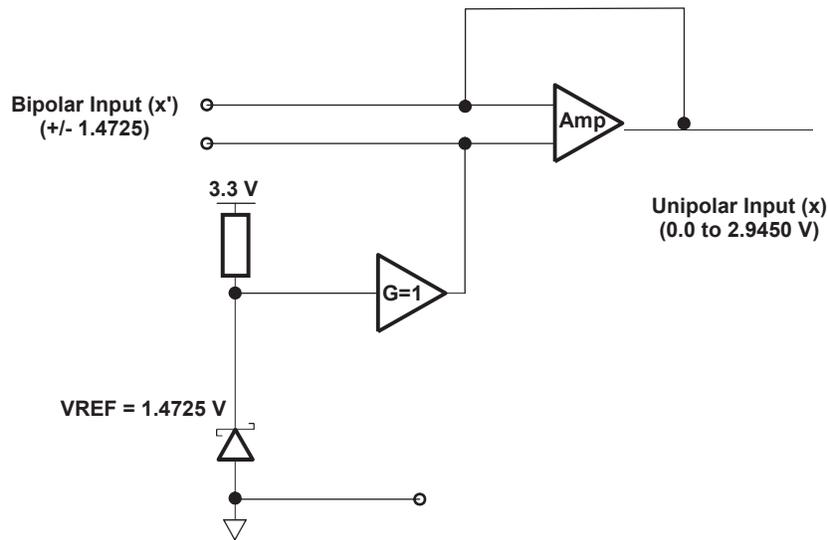


Figure 3. Modified Circuit to Scale Input to Safe Range of ADC

The frame of reference for the input offset error has changed. The offset error is measured relative to the bipolar input when the input value is zero ($x' = 0$). This corresponds to a unipolar input value of $x = 1.4300$ V. The inherent ADC gain and offset errors tend to magnify the error relative to the ideal value.

Calibration

For example, if you assume the ADC has a + 1.5% gain error and + 1.5% offset error, then the bipolar offset error count is as follows:

Bipolar Input:	$x' = 0.0000\text{V}$
Unipolar ADC Input:	$x = 1.4300\text{V}$
Expected Count:	$y_e = 1.4300 \times 4095/3 = 1952$
Actual Count:	$y_a = 1952 \times 1.015 + 60 = 2041$
Bipolar Offset Error:	$y_a - y_e = 2041 - 1952 = 89 \text{ counts (4.6\% error)}$

This is a higher error count than expected, but it can be removed by using calibration.

4 Calibration

Calibration is performed by feeding two known reference values into two ADC channels and calculating a calibration gain and offset to compensate for the input readings from the other channels. This is possible because the channel-to-channel errors are small. The achievable accuracy using calibration is largely dependant on the accuracy of the known references fed into the ADC. The best possible accuracy achievable is limited by the channel-to-channel gain and offset errors of the ADC.

Note: TMS typical channel-to-channel gain and offset errors of $\pm 0.1\%$ have been observed.

The equations to measure the ADC actual gain and offset and calculate the calibration gain and offset are derived as follows.

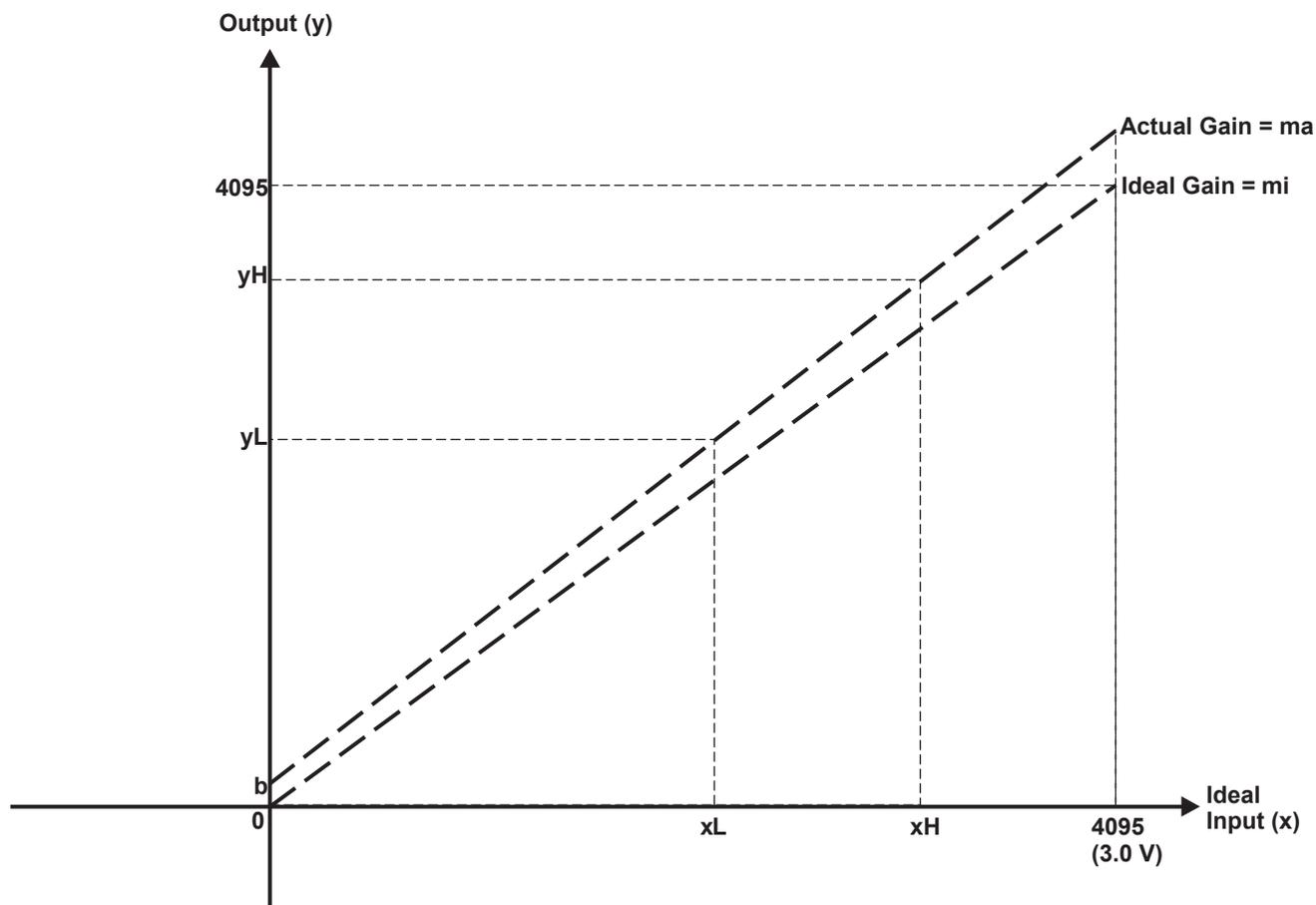


Figure 4. ADC Actual Gain

First, review [Equation 2](#):

$$y = (x \times ma) + b$$

ma = actual gain
 b = actual offset (relative to 0 input)

Now, to derive the actual gain, reference the two converted analog inputs from the graph as follows:

$$ma = \frac{(yH - yL)}{(xH - xL)}$$

xL = known reference low input
 xH = known reference high input
 yL = reference low ADC output
 yH = reference high ADC output

(3)

Derive the actual offset by substituting one of the x/y pairs from the graph and using the gain from [Equation 3](#):

$$b = yL - (xL \times ma)$$

(4)

As you already know the value of y, you need to solve for the value of x by inverting the input and output of [Equation 2](#). To do so, you must rearrange the equation to solve for x as follows. This will give you the calibration of the ADC.

$$x = \frac{(y - b)}{ma} = \frac{y}{ma} - \frac{b}{ma} = (y \times \text{CalGain}) - \text{CalOffset}$$

(5)

The above derivation renames the variables 1/ma and b/ma to CalGain and CalOffset, respectively, so the next step is to solve for those variables as follows in [Equation 6](#) (ADC calibration gain) and [Equation 7](#) (ADC calibration offset).

$$\text{CalGain} = \frac{1}{ma} = \frac{(xH - xL)}{(yH - yL)}$$

(6)

$$\text{CalOffset} = \frac{b}{ma} = \frac{(yL - (xL \times ma))}{ma} = \frac{yL}{(ma - xL)} = (yL \times \text{CalGain}) - xL$$

(7)

In summary, using two known references (xL, yL) and (xH, yH), you can calculate the actual offset and gain errors and calculate the calibration gain and offset using the following formulas.

[Equation 2](#) $y = (x \times ma) + b$ ADC actual equation
[Equation 3](#) ADC actual gain

$$ma = \frac{(yH - yL)}{(xH - xL)}$$

xL = known reference low input
 xH = known reference high input
 yL = reference low ADC output
 yH = reference high ADC output

[Equation 4](#) $b = yL - (xL \times ma)$ ADC actual offset

[Equation 5](#) $x = (y \times \text{CalGain}) - \text{CalOffset}$ ADC calibration equation

[Equation 6](#) $\text{CalGain} = \frac{(xH - xL)}{(yH - yL)}$ ADC calibration gain

[Equation 7](#) $\text{CalOffset} = (yL \times \text{CalGain}) - xL$ ADC calibration offset

The calibration process involves the following five basic steps:

1. Read the reference connected to ADCLO, set the offset to + 10 LSBs (only once from boot).
2. Read the known reference values input channels (yL and yH).
3. Calculate the calibration gain (CalGain) using [Equation 6](#).
4. Calculate the calibration offset (CalOffset) using [Equation 7](#).
5. Cycle through all channels applying the calibration [Equation 5](#).

5 Hardware Connectivity

Calibration requires that two ADC channels be dedicated to supply two known reference inputs, leaving 14 user channels. [Figure 5](#) shows the recommended connection.

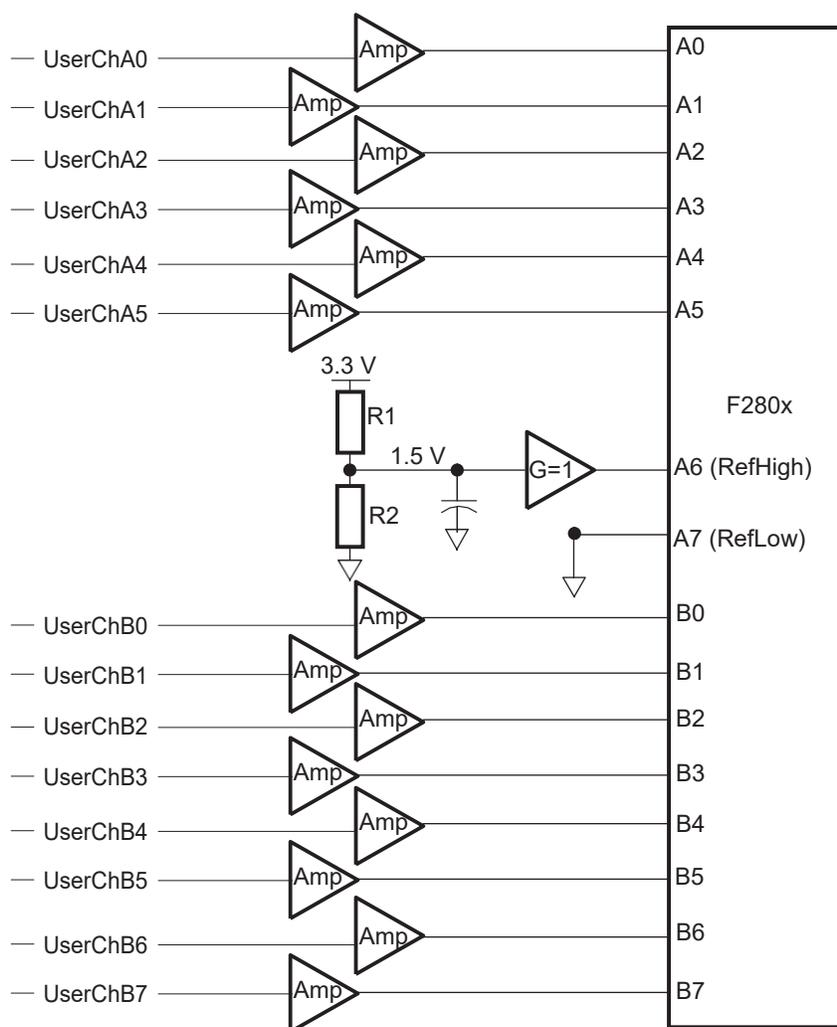


Figure 5. Channels for Calibration, 14 User Channels

Note: The typical channel-to-channel error is approximately $\pm 0.1\%$ across all channels. Therefore, you can choose the channels for calibration, but it is preferable for them to be in the same group.

You can also choose the calibration inputs, based on what is available in the system. Analog ground (ADCLO) was chosen for this example (and the associated example code), because it is readily available in all systems. Use of this as a calibration input allows offset correction with one channel. 1.5V is used as the other calibration input, because it is half the input range to the DSP, and would be present in a bipolar input implementation.

Different reference values can be used depending on availability in the system. However, OFFTRIM needs to be loaded after an initial conversion, in which gain and offset are derived from a two point formula, before a regular system operation can begin.

To retain 16 user channels, set up the system as shown in Figure 6. In this scenario, an external analog switch is added that expands the user channels to 16 and is controlled by a GPIO pin using software. In a simple software implementation, the multiplexed channels are sampled on every alternate cycle, relative to the non-multiplexed channels. This means that the multiplexed channels should be used for slower or supervisory functions. This system leaves you with 6 channel pairs (if using simultaneous sampling mode) that can be used for critical functions.

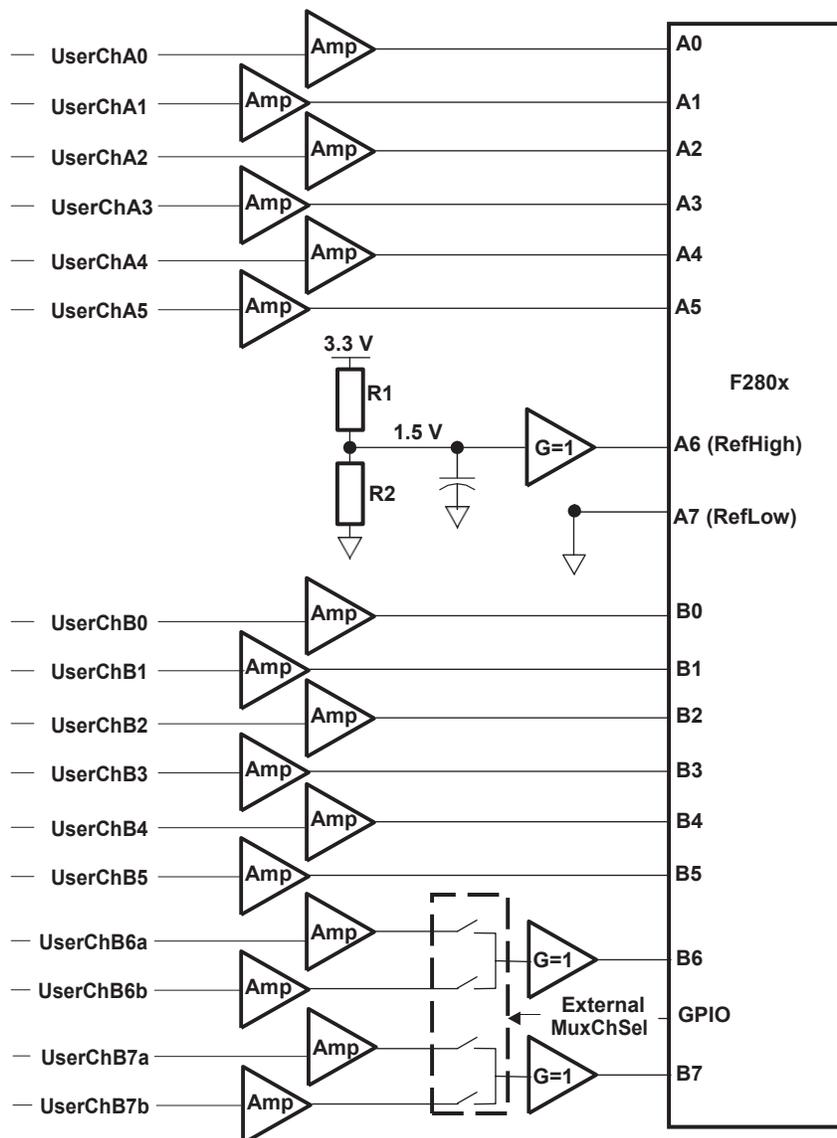


Figure 6. 2 Channels for Calibration, 16 User Channels

Note: A buffer is required at the output of the mux (or any high resistance source) to prevent errors due to high source impedance when sampling the ADC channels.

6 ADC Sampling Techniques

6.1 Sequential Sampling Mode

The ADC converter on the F280x device can operate in sequential sampling mode or simultaneous sampling mode. In sequential sampling mode, the ADC samples one channel at a time and then the sampled signal is passed through four stages of the ADC pipeline for conversion.

Figure 7 shows the timing details of a 16 channel conversion (A0-A7 and B0-B7) in sequential mode based on an event trigger:

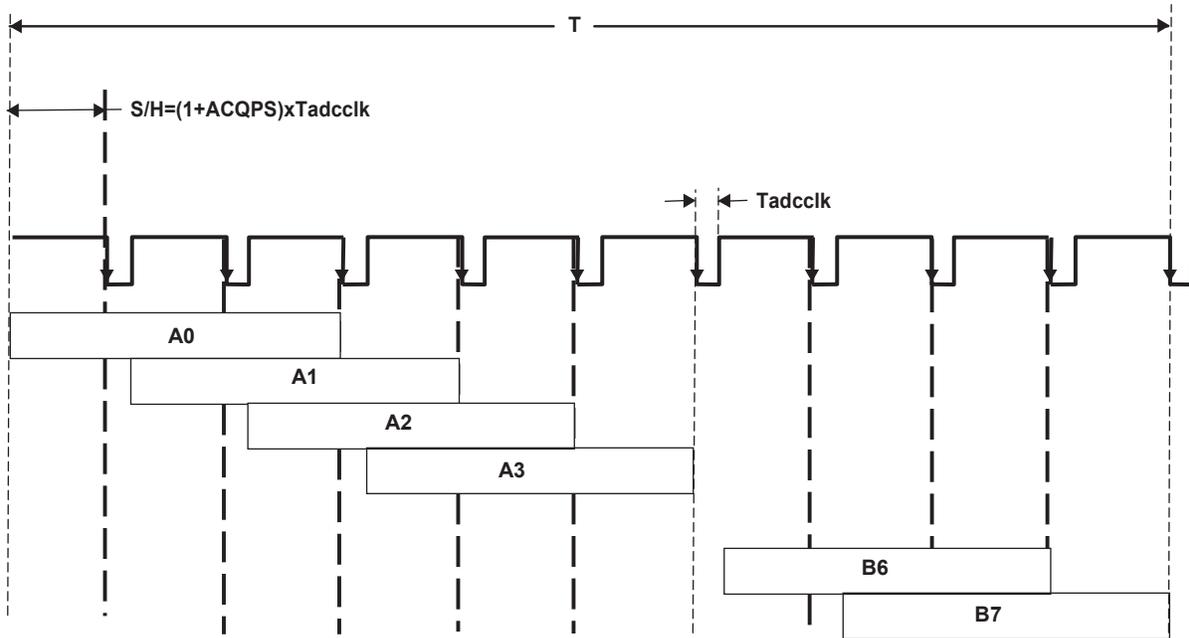


Figure 7. Timing Details of 16 Channel Conversion

Total time for converting 16 channels in sequential mode:

$$T = 17 \times T_{\text{adcclk}} + 18 \times (1 + \text{ACQPS}) \times T_{\text{adcclk}}$$

Table 2 shows the conversion time required to convert all 16 channels in sequential sampling mode under different ADC clock frequencies and S/H windows:

Table 2. Conversion Time of All 16 Channels

ACQPS ⁽¹⁾	Number of Tadcclk Periods	T in μs (ADCCLK = 25 MHz)	T in μs (ADCCLK = 12.5 MHz)	T in μs (ADCCLK = 7.5 MHz)
0	35	1.40	2.80	4.67
3	89	3.56	7.12	11.87
7	161	6.44	12.88	21.47
11	233	9.32	18.64	31.07
15	305	12.20	24.40	40.67

⁽¹⁾ ACQSPS is the acquisition window width. Value of 0 is equal to one ADCCLK period.

In sequential sampling mode, the ADC can be configured in cascade mode or dual sequencer mode. In cascade mode, 16 ADC conversions can be scheduled sequentially based on an event trigger. The order in which the channels are converted and stored in the result register is controlled by the ADC Channel Selection Control Registers (CHSELSEQ1, CHSELSEQ2, CHSELSEQ3, CHSELSEQ4). For direct mapping of channels to the corresponding result register, program the channel selection control register to the values shown in [Figure 8](#).

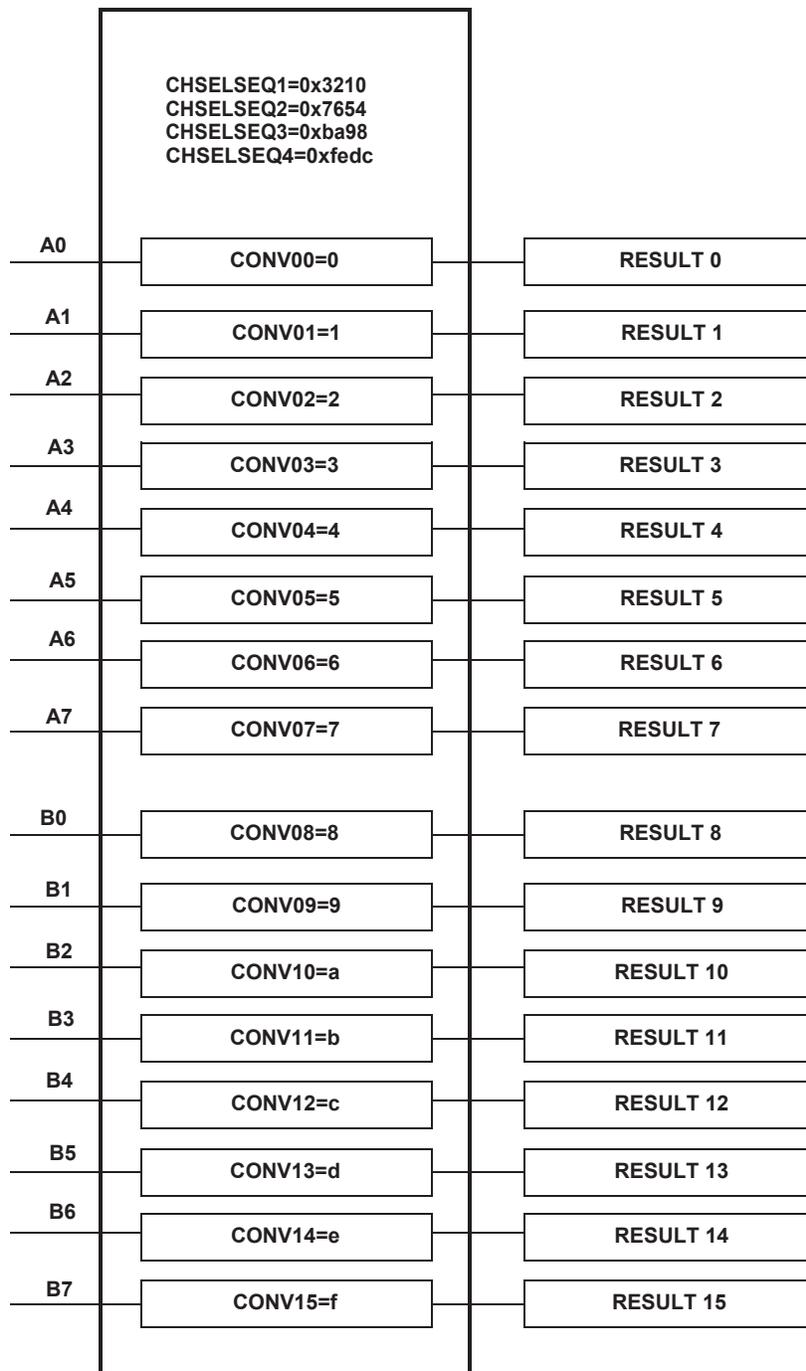


Figure 8. Channel Selection Controls

6.2 Simultaneous Sampling Mode

In simultaneous sampling mode, the ADC can convert input signals on any pair of channels (A0/B0 to A7/B7). Basically, two channels are sampled simultaneously and passed through four stages of the ADC pipeline for conversion. Figure 9 shows the timing details of 8 pairs of channels (A0/B0 to A7/B7) converted in simultaneous sampling mode based on an event trigger.

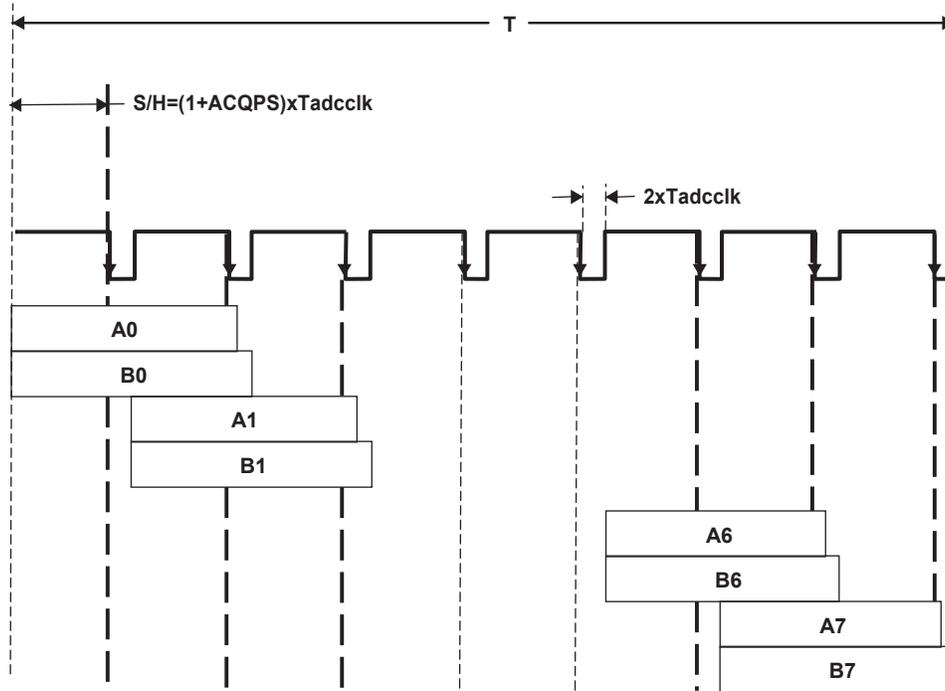


Figure 9. Timing Details of 8 Pairs of Channels

The total time for converting 16 channels in simultaneous mode:

$$T = 9 \times 2 \times T_{\text{adcclk}} + 9 \times (1 + \text{ACQPS}) \times T_{\text{adcclk}}$$

Table 3 shows the conversion time required to convert all 16 channels in simultaneous sampling mode under different ADC clock frequencies and S/H windows.

Table 3. Converting 16 Channels in Simultaneous Mode

ACQPS	Number of Tadcclk Periods	T in μs (ADCCLK = 25 MHz)	T in μs (ADCCLK = 12.5 MHz)	T in μs (ADCCLK = 7.5 MHz)
0	27	1.08	2.16	3.60
3	54	2.16	4.32	7.20
7	90	3.60	7.20	12.00
11	126	5.04	10.08	16.80
15	162	6.48	12.96	21.60

In simultaneous sampling mode, you can schedule 8 pairs of simultaneous ADC conversions based on an event trigger (A0/B0, A1/B1, ..., A7/B7). The order in which the channel-pairs are converted and stored in the result register is controlled by the ADC Channel Selection Control Registers (CHSELSEQ1, CHSELSEQ2). To schedule the channels pairs in normal order A0/B0, A1/B1, ..., A7/B7, the channel selection register values must be programmed as shown in Figure 10.

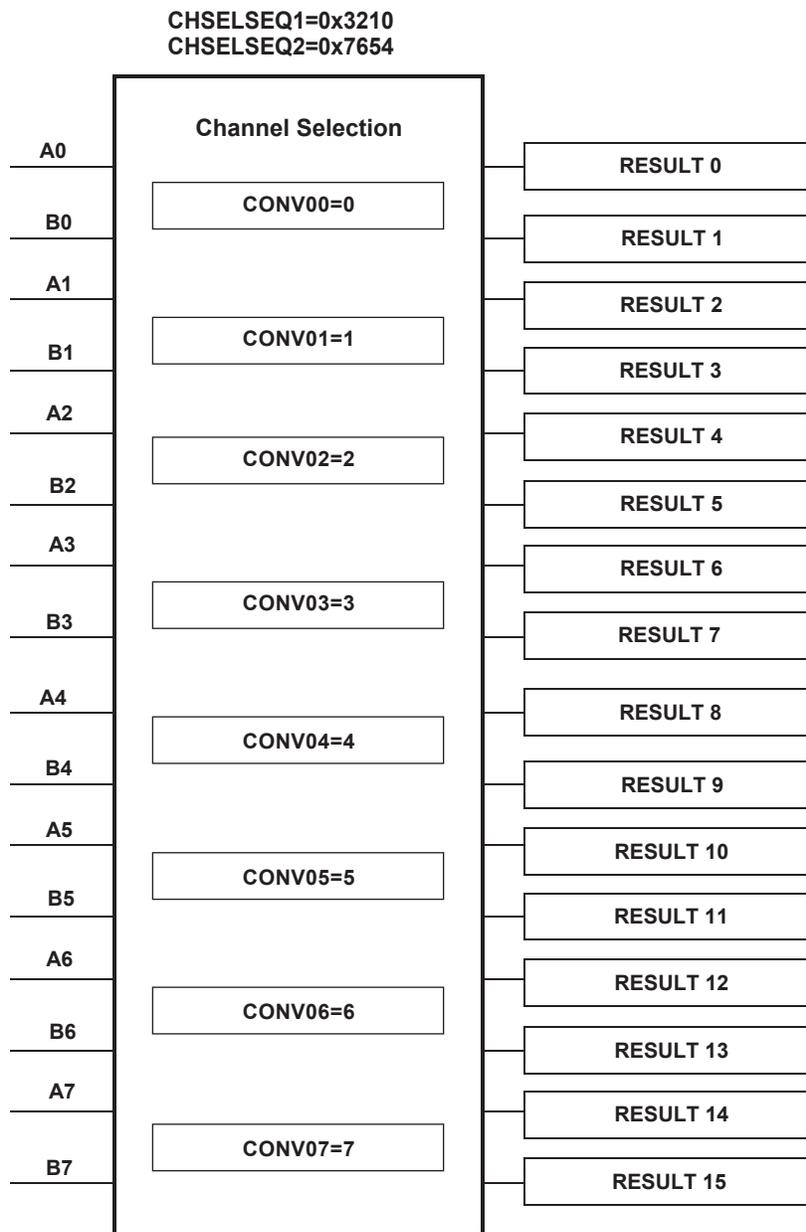


Figure 10. Channel Selection

7 Example Software Calibration Driver

This application report has an option to download an example C program for simultaneous sampling mode and sequential sampling mode that executes from RAM on the F2808 EzDSP. The example program samples ADCLO and sets the offset to + 10 LSBs. The ADC is then configured to process all 16 input channels and generate an interrupt. In the interrupt service routine, a call is made to an optimized assembly driver that reads the user selected ADC channel references, calculates the calibration gain and offset, calibrates all other channels, and stores the information in a RAM structure.

The program supports configuring the ADC for simultaneous or sequential conversion modes:

SEQUENTIAL: ADC channels are converted one at a time:
 A0 -> A1 -> A2 -> ... B0 -> B1 -> B2 ->

SIMULTANEOUS: ADC channels are converted in pairs:
 A0,B0 -> A1,B1 -> A2,B2 ->

The calibrated and converted channels are stored in a RAM structure that contains the following information:

```

typedef struct {
    Uint16 *RefHighChAddr;           // Channel Address of RefHigh
    Uint16 *RefLowChAddr;           // Channel Address of RefLow
    Uint16 *Ch0Addr;                // Channel 0 Address
    Uint16 Avg_RefHighActualCount;   // Ideal RefHigh Count (Q0)
    Uint16 Avg_RefLowActualCount;    // Ideal RefLow Count (Q0)
    Uint16 RefHighIdealCount;       // Ideal RefHigh Count (Q0)
    Uint16 RefLowIdealCount;        // Ideal RefLow Count (Q0)
    Uint16 CalGain;                 // Calibration Gain (Q12)
    Uint16 CalOffset;               // Calibration Offset (Q0)
    // Store Calibrated ADC Data (Q0):
    // Simultaneous Sequential
    // =====
    Uint16 ch0;                      // A0 A0
    Uint16 ch1;                      // B0 A1
    Uint16 ch2;                      // A1 A2
    Uint16 ch3;                      // B1 A3
    Uint16 ch4;                      // A2 A4
    Uint16 ch5;                      // B2 A5
    Uint16 ch6;                      // A3 A6
    Uint16 ch7;                      // B3 A7
    Uint16 ch8;                      // A4 B0
    Uint16 ch9;                      // B4 B1
    Uint16 ch10;                     // A5 B2
    Uint16 ch11;                     // B5 B3
    Uint16 ch12;                     // A6 B4
    Uint16 ch13;                     // B6 B5
    Uint16 ch14;                     // A7 B6
    Uint16 ch15;                     // B7 B7
    Uint16 StatusExtMux;             // Indicates Status Of External Mux For
    // Current Conversion
}ADC_CALIBRATION_DRIVER_VARS;
  
```

This program also supports a GPIO pin toggle for activating an external analog mux to expand the usable channels.

To adapt to the system needs, configure the assembly time switches and settings contained in the header file:

F280x_ADCcalibrationDriver.h

Example Software Calibration Driver

You must either select the simultaneous or sequential sampling mode of operation. For example:

```
#define          SEQUENTIAL                1
#define          SIMULTANEOUS              0
#define          ADC_SAMPLING_MODE        SIMULTANEOUS
```

You must also select which ADC channels are connected to reference high and reference low and the ideal count value. For example:

A6 = RefHigh = 1.5 V ($1.5 \times 4095 / 3.0 = 2048$ ideal count)

A7 = RefLow = 0 V ($0 \times 4095 / 3.0 = 0$ ideal count)

```
#define          REF_HIGH_CH                A6
#define          REL_LOW_CH                 A7
#define          REF_HIGH_IDEAL_COUNT      2048
#define          REF_LOW_IDEAL_COUNT       0
```

The number of cycles required to execute the calibration driver is:

109 cycles or 1.09 μ S @ 100 MHz (6.75 cycles per user channel)

Without calibration, the driver would take approximately 2.4 cycles per user channel to read and store the ADC input. The calibration overhead is therefore approximately 4.3 additional cycles per channel.

Analog offset correction is performed once at start up to load the OFFTRIM register with the correction value. This function takes approximately 120 CPU cycles at maximum ADC speed for code execution. The sampling speed of the ADC adds to this cycle count because a conversion of 8 samples is initiated by this routine.

Note: In the example C program, the gain and error calculations are performed on every ADC interrupt. To reduce the overhead, these calculations can be performed in the background at regular intervals, as seen in the example program.

Only the running weighted average of the reference inputs should be retained in the interrupt routine. The calculation of CalGain and CalOffset can be pushed to the background. This can save up to 30 cycles per interrupt.

If further performance optimization is required, the calibration driver can be inlined instead of making it a C function call.

8 Gain and Offset Temperature Drift

Once initial calibration values for gain and offset have been determined, CPU cycles can be saved by reusing these values, rather than calculating them every time an ADC conversion is performed. The frequency in which the application updates these values is determined by the stability of the input references, input drivers, and the ADC itself over temperature. This section allows you to determine how often, if at all, calibration values need to be updated based on gain and offset changes internal to the ADC.

Gain Error Drift

The change in gain error over temperature is determined by the movement of the internal or external reference. For the purposes of this document, only the internal reference is considered, as the movement of an external reference over temperature is unique to the source used to supply the reference.

The TMS320280x datasheet defines the temperature coefficient of the internal reference as 50 ppm/°C. Using Equation 8, for a Q-rated device (-40°C to 125°C), this would equate to a temperature shift of:

$$\text{Gain Shift} = ((\text{Temperature Movement } ^\circ\text{C}) \times 50 \text{ ppm } / ^\circ\text{C}) / 1000000 \quad (8)$$

$(165^\circ\text{C} \times 50 \text{ ppm}/^\circ\text{C}) / 1000000 = 0.00825$ or 0.825% gain shift.

This formula can be used to calculate the expected shift in gain error over any temperature delta the device may undergo in the system. The maximum shift described above is not likely to be observed in most systems. For example, the original calibration point may be at room temperature and movement will be a \pm percentage scaled by the temperature difference about that origin.

The direction of the gain error shift is determined by the direction of the temperature movement. If temperature is increasing, the gain error increases by the amount in Equation 8. If the temperature is decreasing, the gain error decreases by the amount in Equation 8.

Figure 11 illustrates the full Q-rated temperature movement's effect on gain error:

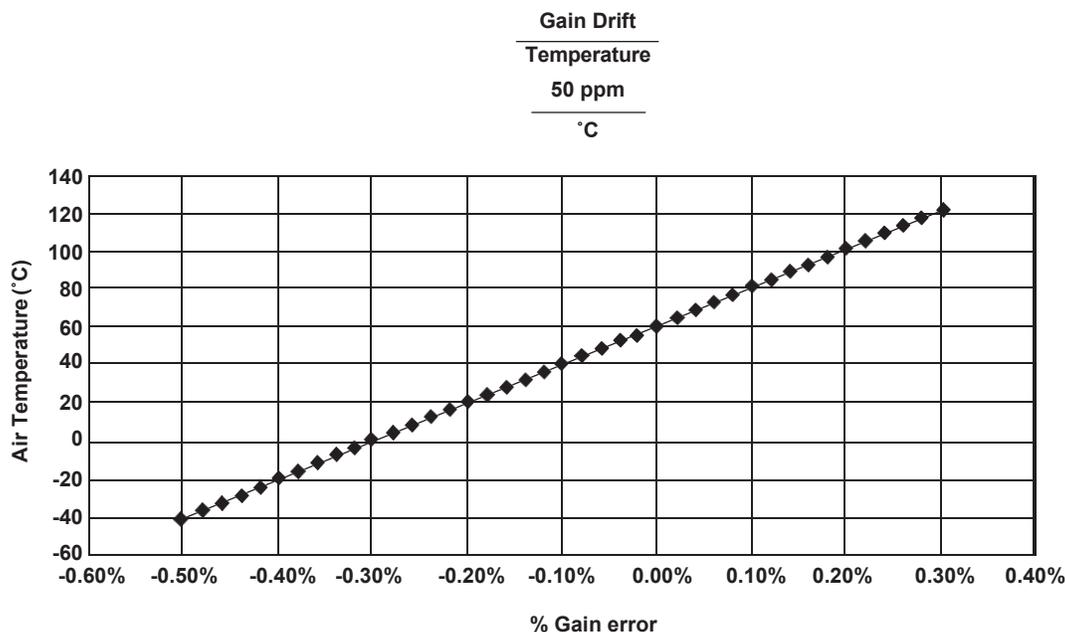


Figure 11. Gain Drift Over Temperature

Offset Error Drift

Offset error drift over temperature is typically 4 LSBs for the full temperature range of a Q-rated device (-40°C – 125°C). This drift is measured end point to end point, so total drift from a center point would be ± 2 LSBs. If the end application temperature drift is less than listed below, the associated offset drift scales accordingly. Note that if the calibration driver is not recalled after a temperature movement of the device, the offset correction could be too large. For low input voltages, this might create an underflow condition, resulting in a calibrated ADC value of 0xFFFF.

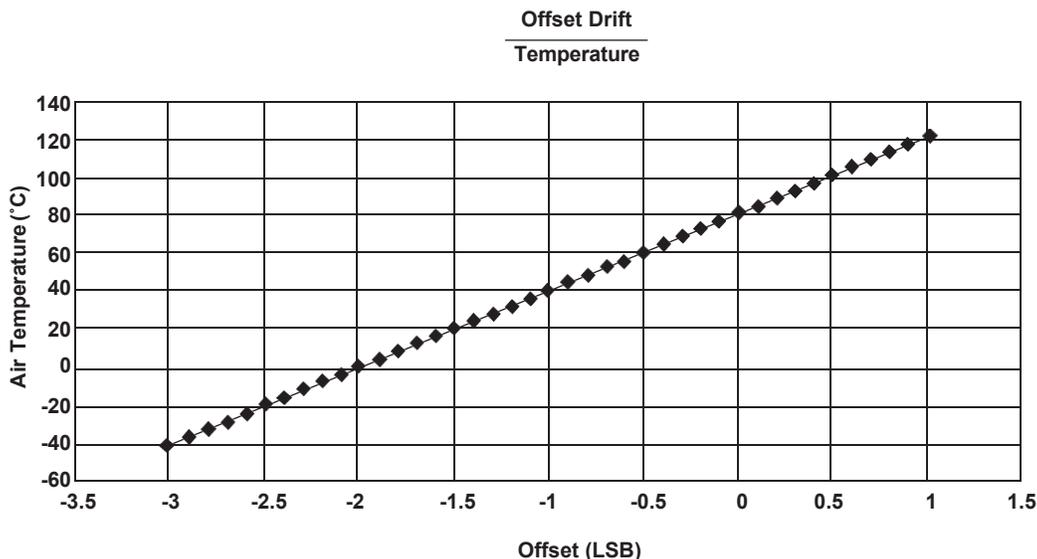


Figure 12. Offset Drift Over Temperature

9 Useful Tips

The following list includes tips or options to improve or assure accuracy:

- **Provide a low resistance path for the ADCLO pin:** Always make sure that the ADCLO pin on the F280x device is connected directly to analog ground. Any resistance on this pin degrades the offset error, as well as the stability of the conversion.
- **Use low ESR capacitors for the ADCREFP/ADCREFM pins:** Follow the datasheet recommended specification for ESR requirements on these pins. Historically, this has required the use of ceramic capacitors, because electrolytic capacitors typically have a large ESR at the ADC clock rate. For the same reason, loading of these pins is not recommended.
- **Use the bipolar input conversion reference as calibration input:** When converting bipolar to unipolar signals, make sure that the reference voltage used as the *midpoint* of the ADC range is fed in as an input calibration channel. This removes any bipolar offset error (as discussed earlier in this application report).
- **Digital and analog grounds connected at one point:** To avoid noise created by digital current loops, connect the digital and analog grounds at one point, making sure that any digital or analog current loops do not cross through this point. This is common practice when mixing digital and analog on a single board or device.
- **Correcting for off chip offset:** Depending on the external components used, there is potential for offset added to the input signal caused by the input drivers. To correct for this offset, as well as the offset of the 280x device, you can run the calibration routine with two known channels and then load the OFFTRIM register with the inverse of the calculated offset before system conversions take place. In this case, the OfftrimInit function can be omitted. Off chip gain continues to be corrected inline to the calibration driver.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2019, Texas Instruments Incorporated