

ADC Offset in MSC12xx Devices

Michael Gurevich

Data Acquisition Products

ABSTRACT

The offset parameter plays a significant role in analog-to-digital (A/D) conversion, especially in the high-resolution delta-sigma A/D converter implemented in the MSC1210, MSC1211 and MSC1212 devices. This application report discusses the following points:

- a) The main sources of ADC offset in these devices;
- b) Compensating the offset by using either hardware or software;
- c) The Null_Offset program, which gives the user an opportunity to try different kinds of offset compensation.

This report refers to these three devices as the MSC12xx devices. The information contained in this bulletin also applies to the ADS1216, ADS1217 and ADS1218 devices.

Contents

1	ADC Structure in the MSC12xx Devices	3
2	Standard ADC Calibration Procedures	4
3	What Affects ADC Offset?	5
	3.2 A/D Conversion Formula	13
4	Offset Compensation Methods	14
	4.2 Device Offset Calibration Commands	14
	4.3 Using Offset DAC	15
	4.4 Offset Compensation by Swapping Inputs	16
	4.5 Offset Subtraction from ADC Result.....	16
	4.6 Placing Necessary Offset Value in OC Register	17
	4.7 Calculation During Program Design State.....	17
5	The Null_Offset Program	19
6	Recommendations	20
	References.....	20

Figures

Figure 1.	MSC121xx ADC structure.....	4
Figure 2.	ADC Offset from PGA Gain and Temperature, Buffer On	6
Figure 3.	ADC Offset from PGA Gain and Temperature, Buffer Off	6
Figure 4.	ADC Offset after Self-Calibration, from PGA Gain and Temperature, Buffer On	7
Figure 5.	ADC Offset after Self-Calibration, from PGA Gain and Temperature, Buffer Off.....	7
Figure 6.	ADC Offset from Input Pins, Buffer ON.....	9
Figure 7.	ADC Offset from Input Pins, Buffer OFF.....	9
Figure 8.	ADC Data, PGA Gain = 128.....	10
Figure 9.	ADC CMRR from PGA Gain, Buffer On	11
Figure 10.	ADC CMRR from PGA Gain, Buffer Off.....	11
Figure 11.	ADC PSRR from PGA Gain.....	12
Figure 12.	Offset DAC INL	15

Tables

Table 1.	ADC Sampling Frequency from the PGA Gain.....	4
Table 2.	Digital Filter Bit Shift Gain Value and Decimation Ratio	13
Table 3.	ODAC codes	16

1 ADC Structure in the MSC12xx Devices

The ADC structure of the MSC12xx is shown in Figure 1.

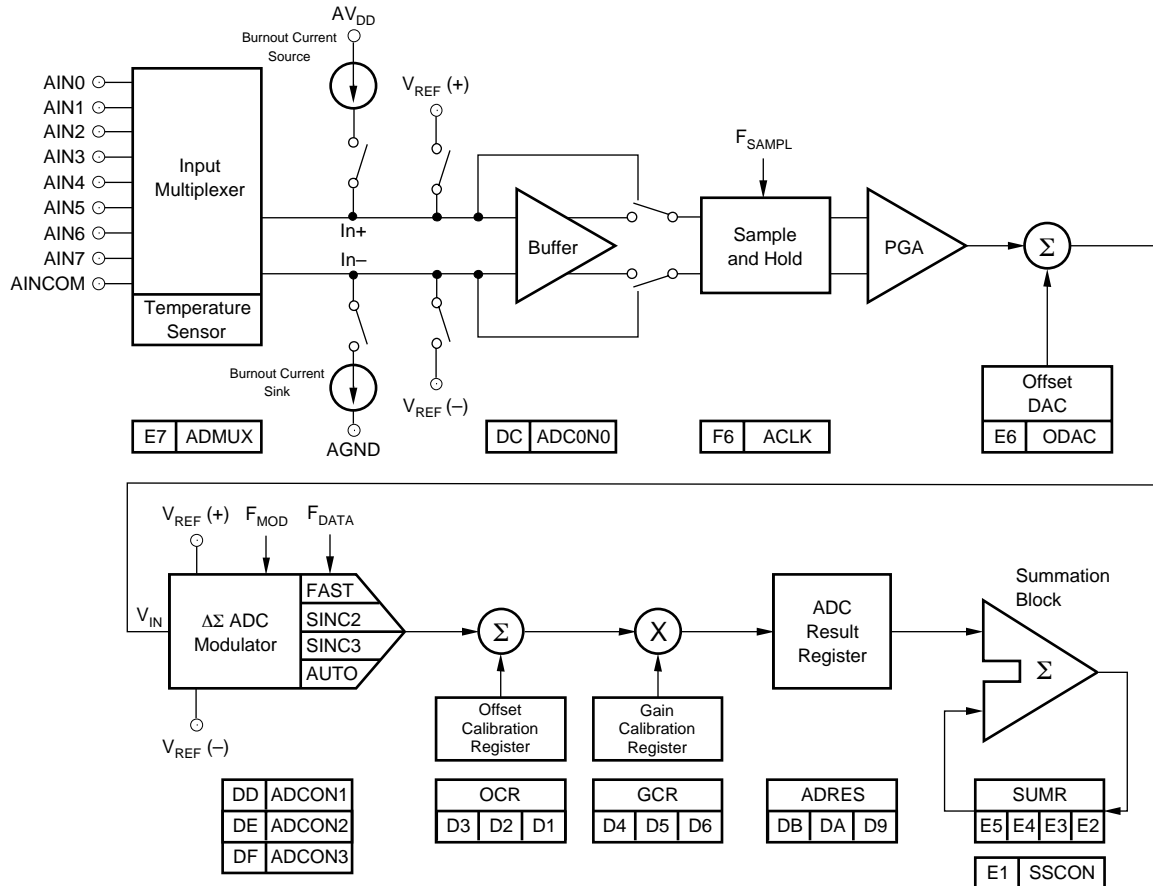


Figure 1. MSC12xx ADC structure

The main block in the ADC is a delta-sigma ($\Delta\Sigma$) modulator, which includes the comparator, integrator and digital filters. The modulator clock frequency is:

$$F_{MOD} = F_{CLK} / (ACLK + 1) / 64$$

where F_{CLK} is a processor clocking frequency and ACLK is the content of the ACLK SFR register (Special Function Register). The ADC data comes from the delta-sigma modulator output. The ADC data rate is defined as:

$$F_{DATA} = F_{MOD} / DEC$$

where DEC is the ADC decimation ratio, which is the number of modulator outputs that are used to compute one output data sample. The sampling frequency and sampling capacitor depend on the Programmable Gain Amplifier (PGA) gain and modulator clock. Table 1 shows the sampling frequency values, sampling capacitor values, and corresponding PGA gain settings for the MSC12xx.

Table 1. ADC Sampling Frequency from the PGA Gain

PGA	F _{SAMPL}	C _{SAMPL} (pF)
1	F _{MOD}	9
2	F _{MOD}	18
4	F _{MOD}	36
8	F _{MOD} X 2	36
16	F _{MOD} X 4	36
32	F _{MOD} X 8	36
64	F _{MOD} X 16	36
128	F _{MOD} X 16	36

From the modulator, the ADC data comes to the offset compensation block, then to the gain compensation stage. The resulting 24 bits of data are available at the ADRES registers. The data can also be forwarded to the 32-bit summation register SUMR, where ADC result averaging can be executed by device hardware.

2 Standard ADC Calibration Procedures

MSC12xx ADC calibration should be performed each time after any one of the following change: the device powers ON; a significant change in the operating temperature; changes in the decimation; PGA gain; input buffer switch, or a significant change of power supply voltages. Calibration can be done separately for offset and for gain.

There are five calibration commands available through the ADCON1 register. The first procedure that should be done is offset calibration. When the offset self-calibration command is issued, the ADC inputs are disconnected from the device inputs and both connected to the AINCOM pin. The digital filter is zeroed and the calibration starts. The first three samples are ignored, giving the data time to pass through the filter. Then, an average of the next four samples is used to set the offset calibration registers. Altogether, the offset calibration process requires seven data samples.

To get the best results from offset self-calibration, it is necessary to have the AINCOM pin steady during calibration. It should be connected to the analog ground. To avoid common mode voltage change influence, you can also connect AINCOM to the common voltage of the input differential signal.

The self-gain calibration is made after the offset calibration. This calibration process also requires seven data cycles. The ADC inputs are internally connected to the reference pins; PGA is set to gain one; the first three samples are discarded, and the average of the next four samples is used to set the gain calibration registers. The combined self offset and gain calibration commands will take 14 data cycles before the ADC is ready for use. There is no need to perform gain calibration after each offset calibration.

The system-offset calibration allows you to compensate for an internal and external circuit offset. To perform this, the user applies a steady differential input signal and then starts the system-offset calibration. When the system-offset calibration is finished, the ADC converts this input signal as zero. The system-offset calibration can cover the full input signal range. Before doing either the self-gain or system-offset calibrations, the Offset DAC (digital-to-analog converter) should be zeroed; otherwise, calibration will neutralize the effect of the Offset DAC.

The system gain calibration should be done after system-offset calibration. To do this, the user should apply the desired positive full-scale input signal and begin calibration. With the system gain calibration, you can change the ADC input range from 66% below to 33% above the nominal gain V_{REF}/PGA . Each of the system calibrations takes seven data samples as described earlier.

If you switch from one ADC setting to another, it is not necessary to perform a calibration each time. You can store the content of the calibration registers for each configuration and reload the registers when you change ADC settings.

3 What Affects ADC Offset?

1) PGA gain and temperature change. When the PGA is changed, the absolute value of the ADC offset stays the same; on a relative scale, however, the offset changes significantly. Figures 2, 3, 4 and 5 demonstrate the ADC offset as calculated to an effective input signal; that is, to find the actual ADC offset, simply multiply the chart value by the PGA gain.

Figure 2 shows that when the buffer is used, the offset does not change significantly over the temperature range; thus, there is no need for recalibration. It can also be seen that gains 16, 32, 64 and 128 have almost the same offset, and switching between them allows the use of the same offset calibration. In contrast, gains 1, 2, 4 and 8 need a unique offset calibration in most cases.

Figure 3 indicates that the ADC offset changes when the buffer is OFF. Over the full operating temperature range for gains 1, 2, 4 and 8, the offset value changes between 10 and 20ppm. For gains 16, 32, 64 and 128, the offset value shifts between 5 and 8ppm. For some applications, this shift may be not acceptable; such instances would require temperature tracking, because the ADC must be recalibrated whenever the temperature changes.

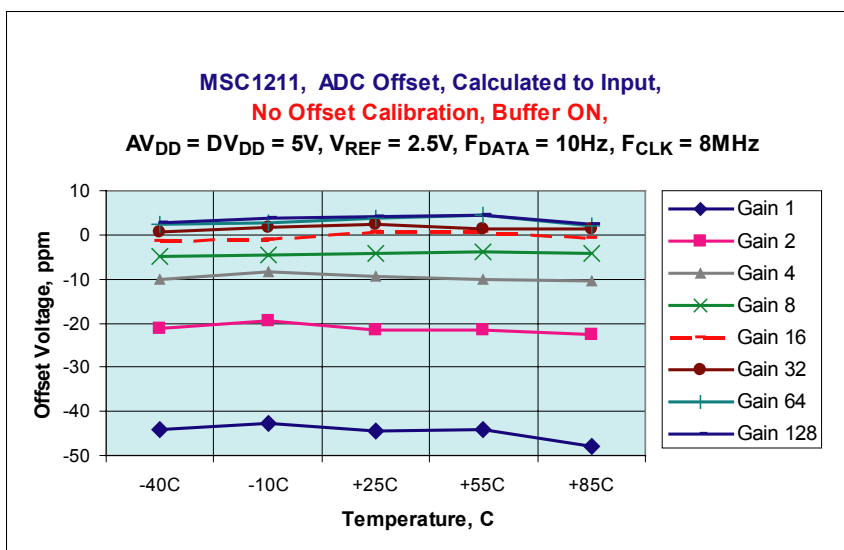


Figure 2. ADC Offset versus PGA Gain and Temperature, Buffer On

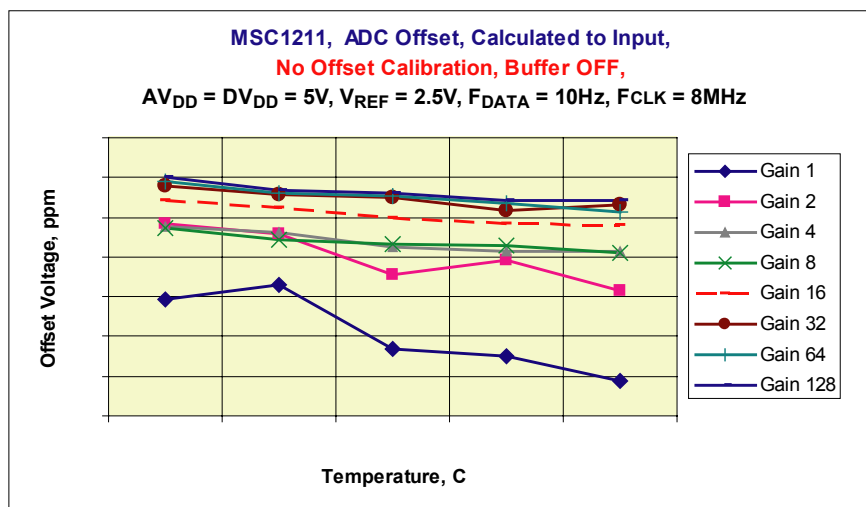


Figure 3. ADC Offset versus PGA Gain and Temperature, Buffer Off

Figure 4 shows how precise the ADC offset self-calibration command works when the buffer is on. For example, for gain 128 in full temperature range, the offset self-calibration error is about 3ppm, which is comparable to the offset temperature drift. In this case, this result means that there is no real reason to perform the offset self-calibration procedure when the device temperature changes. For gain 128, the offset input error of 3ppm gives a possible ADC result error of 384ppm. If this level of offset error is not acceptable, then other methods of offset compensation should be considered.

Figure 5 demonstrates the precision of the ADC offset self-calibration procedure when the buffer is off. The offset self-calibration error is about 20ppm. Where this degree of error is not acceptable, other methods of offset calibration should be applied.

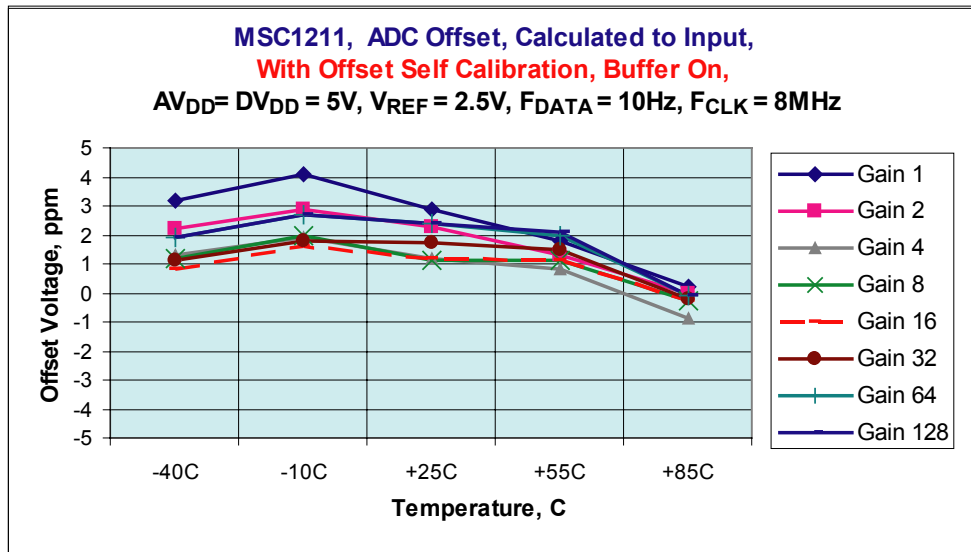


Figure 4. ADC Offset after Self-Calibration, versus PGA Gain and Temperature, Buffer ON

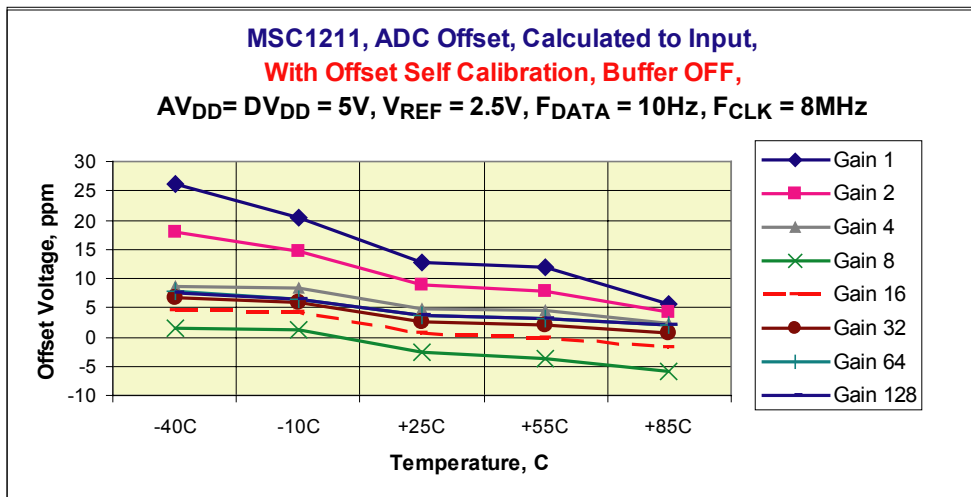


Figure 5. ADC Offset after Self-Calibration, versus PGA Gain and Temperature, Buffer OFF

2) Input buffer. The input buffer minimizes ADC input bias currents, and therefore the voltage drop over input circuit resistance becomes very small. It is important to use a buffer when PGA gain is greater than 8 or the modulation clock is higher than the standard 15.6KHz. In these cases, sampling frequency increases, and leads to a higher input current, which in turn creates a greater offset error and also puts more load on the signal source. Comparing Figs. 2 and 3, note how small the ADC offset and temperature drift can become when the buffer is on. Alternatively, Figs. 4 and 5 show that the self-calibration procedure works much better with the buffer on.

One buffer disadvantage is an increased analog supply current and the reduction of the input signal range. This can be a problem if the user is trying to apply an input signal below $AGND+50mV$ or greater than $AV_{DD}-1.5V$.

3) The input pins pair that is used for A/D conversion also affects the offset. The rule is very simple: the larger the difference between ADC input pin numbers, the greater the offset value. To minimize the offset, try to select neighboring pins; in this case, offset is minimal and has little dependence on the chosen pair. If possible, it is good practice to use two free neighboring pins for system-offset calibration. These two inputs are both connected to the source signal common voltage. System offset calibration (or other methods of calibration, described later) uses this input pair for offset calibration; the program then applies this calibration result to all other neighbor input pairs. As Figures 6 and 7 show, system offset calibration gives much better results than self offset calibration.

Figs. 6 and 7 are only provided as examples, and should not be used to predict precise ADC offsets. Actual offset values can depend on a number of factors, including layout of input circuits and ground configuration.

If the inputs are swapped, the offset changes the sign, but the absolute value remains approximately the same. This feature can be used to calculate offset value out, which is discussed later. Figs. 6 and 7 also show that the input buffer significantly decreases the input pins effect, and offer one more argument for buffer usage. These graphics also give an answer to the question of whether or not it is necessary to make offset calibration when input channels are changed. For example (according to Fig. 6), if the acceptable level of the offset error is 10ppm, and the system is working with the buffer and needs to measure single-ended signals, then the pair $AIN0-AIN1$ will have an offset error of $\pm 3ppm$ and the pair $AIN0-AIN7$ will have an offset error of around 13ppm, which does not satisfy the criteria. A much better result would be if the second pair is $AIN0-AIN2$; then an offset is the same, 3-4ppm. If the situation still forces the use of the pair $AIN0-AIN7$, then a system offset calibration or another calibration method should be applied.

In a case when 8 single-ended signals should be measured, some improvement may be achieved if the pin $AIN4$ is chosen as common; the most remote pair would then be $AIN4-AIN0$ and $AIN4-AINCOM$, which would reduce maximal offset error by a factor of two.

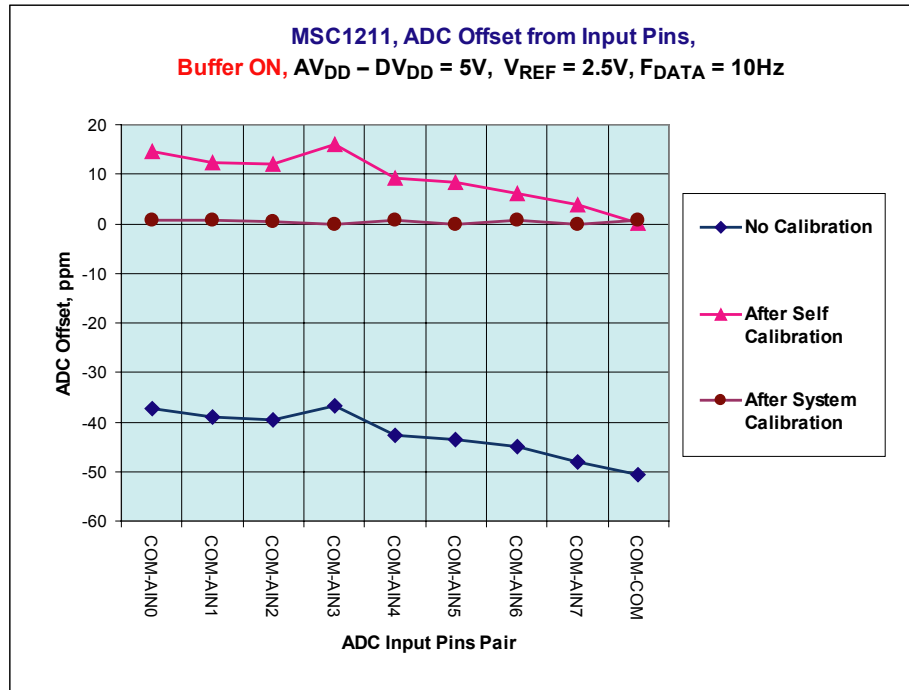


Figure 6. ADC Offset versus Input Pins, Buffer ON

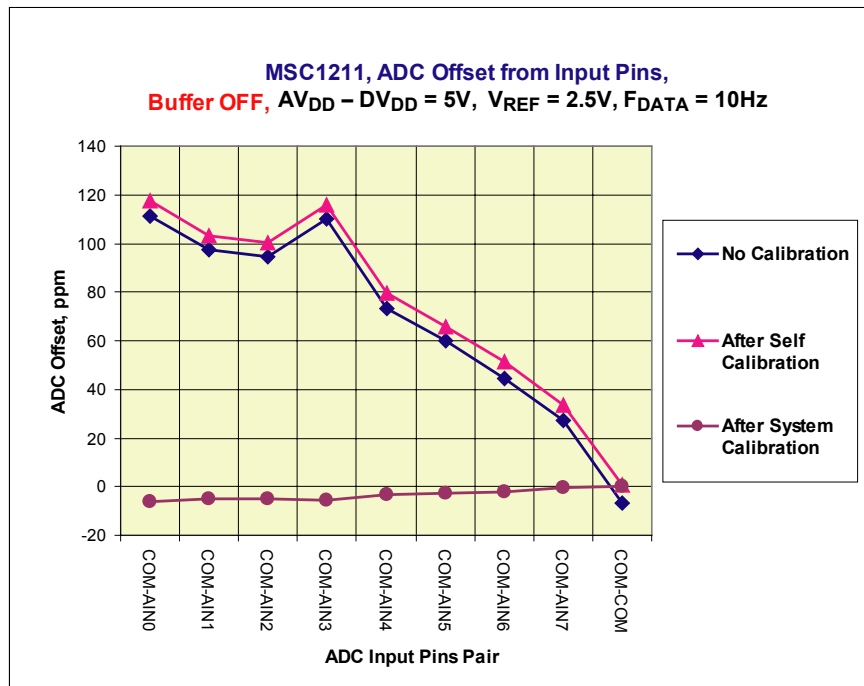


Figure 7. ADC Offset versus Input Pins, Buffer OFF

4) Conversion noise can also significantly affect the offset. The reason for this is that the offset calibration command (or system offset calibration command) uses only 4 samples to establish the average value. Thus, the more noise that exists in the A/D conversion results, the more random the value will be that is placed in the offset calibration register. Figure 8 shows 100 ADC samples taken with $\text{PGA} = 128$, $V_{\text{REF}} = 2.5\text{V}$, $V_{\text{input differential}} = 0\text{V}$, $F_{\text{DATA}} = 10\text{Hz}$ and the resulting ENOB of 18 bits. That is, 4 sequential AD samples will very likely generate a different average from the displayed average of 100 samples. If these 4 samples are used for offset self-calibration, the ADC would still have some offset error.

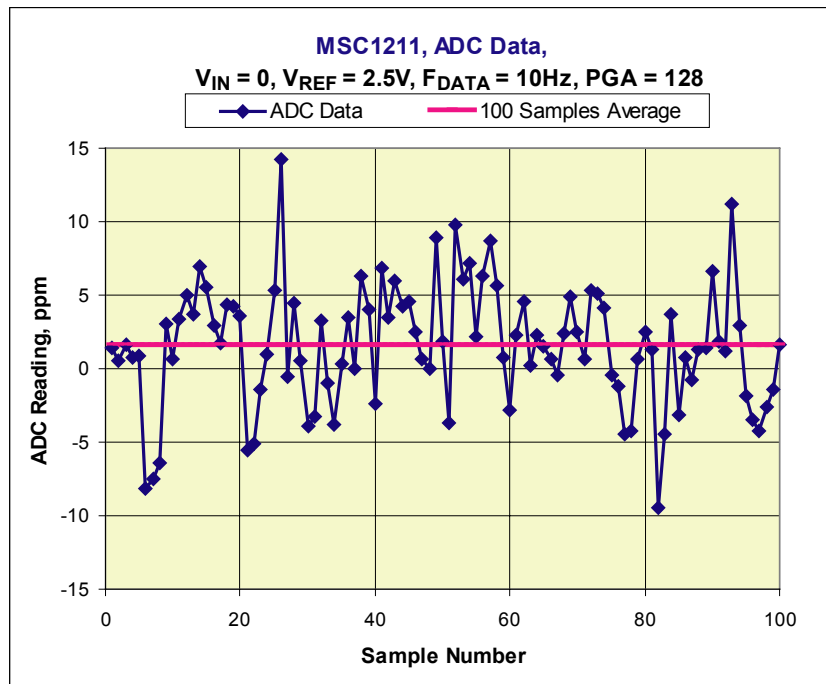


Figure 8. ADC Data, PGA Gain = 128

5) Input common voltage change introduces DC error which can vary depending on the PGA gain. Unlike linear amplifiers, where CMRR error signals tend to be a linear function of gain, errors in the ADC are more a function of the input sampling capacitor array used when implementing PGA functions. Therefore, if you are going to work with high PGA gains and the common voltage of the input differential signal is supposed to change, consider using an external differential buffer that can remove the common voltage change from the input signal. Another consideration is that in a data sheet, CMRR is usually given when the input differential signal is zero. Figures 9 and 10 show typical CMRR curves, and demonstrate how CMRR changes with respect to the PGA when V_{IN} is zero and when V_{IN} is equal to half of the input range. These charts also show how CMRR would change if the common voltage affects only the input stage and the PGA is an ideal device which amplifies the input circuit offset by 6dB (two times) for each gain setting.

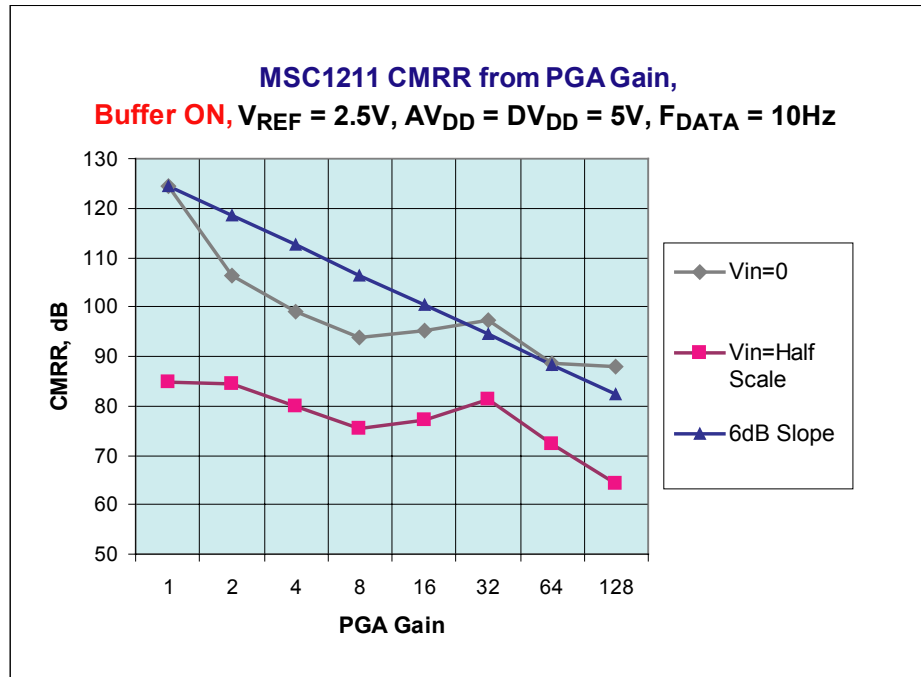


Figure 9. ADC CMRR versus PGA Gain, Buffer ON

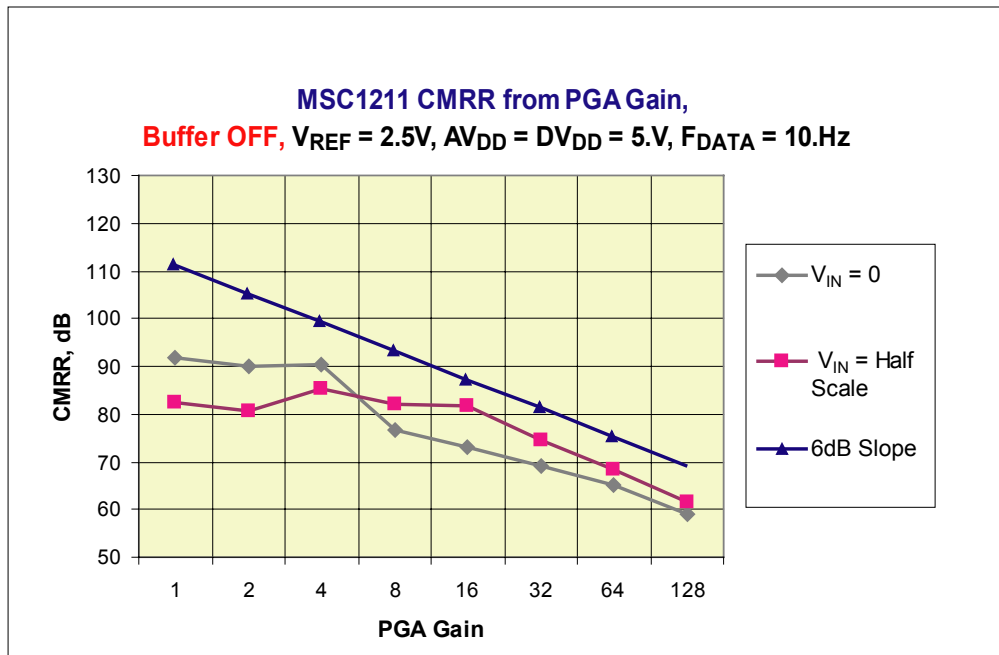


Figure 10. ADC CMRR versus PGA Gain, Buffer OFF

6) Power supply changes affect the ADC offset. The PGA amplifies the offset change; therefore, depending on PGA gain, PSRR has the same dependence as in the CMRR case. Furthermore, this effect means that at high gains, special attention should be applied to stabilize both the analog and digital supplies. If this is impossible, external signal amplification should be considered. Figure 11 shows the typical offset dependence from both power supplies. The third line shows how PSRR decreases from the gain in a case when the PGA is an ideal device.

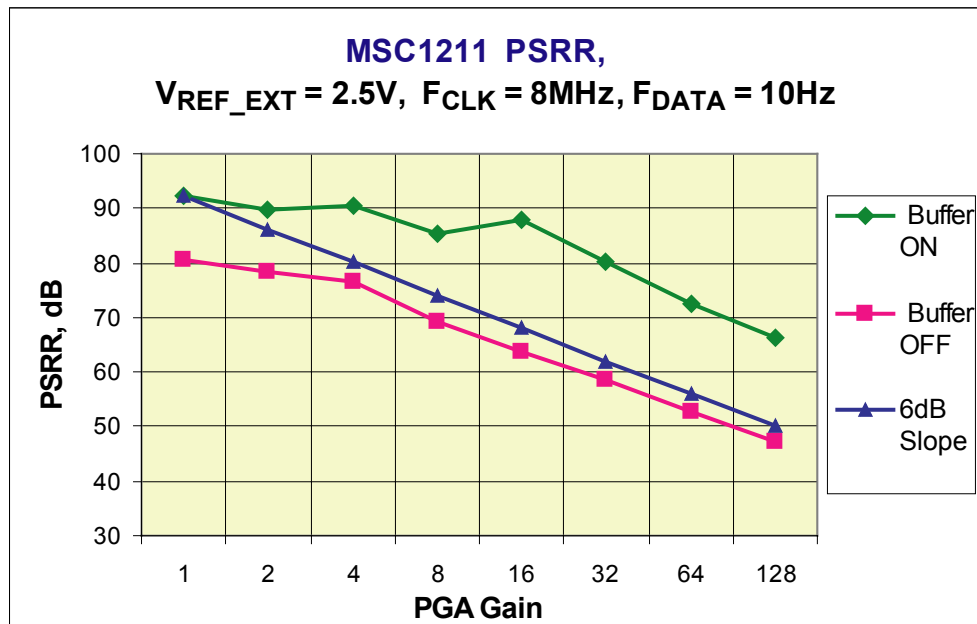


Figure 11. ADC PSRR versus PGA Gain

7) There are several other factors (usually less significant) that can also affect ADC offset, which are not the subject of this article. These factors include:

- Sampling frequency (modulation frequency) change;
- Digital and analog power supply current change;
- ADC reference inputs common mode voltage change (typical value is 110dB);
- Signal source resistance and capacitance.

3.2 A/D Conversion Formula

To better understand how to compensate an offset, it is necessary to know the A/D conversion equation.

$$ADRES = [(V_{IN} / V_{REF}) * 0.75 * BG * DEC^3 - OC] * GC / N \quad (AD)$$

Where:

ADRES is an ADC result, located in SFR registers ADCRESH, ADCRESM, ADCRESL.

V_{REF} is an ADC differential reference voltage, applied between VREFIN+ and VREFIN- pins.

DEC³ is a third power of ADC decimation ratio value, located in registers ADCON2 and ADCON3. The DEC value is usually chosen when the system is designed and does not change when the program is running. DEC must be greater than 8.

BG is a bit shift gain, which depends on decimation ratio and is applied by the ADC hardware. The user has no control over it (see Table 2). To determine the BG value, round the decimation value to the closest larger decimation number, as shown in Table 2.

Table 2. Digital Filter Bit Shift Gain Value and Decimation Ratio

Decimation	BG Value	Decimation	BG Value
20	2 ¹⁰	255	2 ⁻¹
25	2 ⁹	322	2 ⁻²
31	2 ⁸	406	2 ⁻³
40	2 ⁷	511	2 ⁻⁴
50	2 ⁶	645	2 ⁻⁵
63	2 ⁵	812	2 ⁻⁶
80	2 ⁴	1023	2 ⁻⁷
101	2 ³	1290	2 ⁻⁸
127	2 ²	1625	2 ⁻⁹
161	2 ¹	2047	2 ⁻¹⁰
203	2 ⁰		

OC is an offset calibration value, located in registers OCH, OCM, OCL. The data in these registers usually appear as a result of self or system offset calibration.

GC is a gain calibration value, located in registers GCH, GCM, GCL. The data in these registers usually appear as a result of gain calibration.

N is a proportional coefficient. $N = 2^{22}$ for bipolar ADC mode and $N = 2^{21}$ in unipolar mode.

V_{IN} is ADC internal input signal. Its actual value is:

$$V_{IN} = (V_{INp} + V_{INp_error}) * (PGA + PGA_error) + (ODAC + ODAC_error) \quad (IN)$$

Where:

V_{INp} is a differential voltage, applied to the device pins $AIN_x - AIN_y$.

V_{INp_error} is an error, brought to input signal by ADC input circuits, like multiplexer, buffer, comparator and others.

PGA is the gain, defined by contest in ADCON0 zero register. $PGA = 1, 2 \dots 64, 128$.

PGA_error is the PGA gain error, which is small for lower gains, but reaches 3% for gain = 128.

ODAC is an Offset DAC value, which cannot exceed a range ± 0.5 of FSR. When Offset DAC code = 0, Offset DAC is switched OFF, and the **ODAC** and the **ODAC_error** value become zero.

ODAC_error is an offset DAC error. This error is discussed in section 4.2.

It is possible to employ the following offset compensation methods when analyzing the A/D conversion formula:

1. Use device self offset or system offset calibration command.
2. Use offset DAC.
3. Use ADC specific features.
4. Calculate the offset and subtract it from the ADC result.
5. Calculate the offset and put the adjusted value in OC register.

Furthermore, when we discuss offset compensation methods, it could also mean the necessary ADC result shift desired by a user program algorithm. There are two possible scenarios in offset compensation. First, a known signal is applied to the input; the program measures this input and stores the result, and subtracts it from subsequent ADC samples. One example of this approach is the ADC system calibration command. The second scenario occurs when the program uses programmed procedures to minimize the offset. An example of this method is the offset compensation by input swapping.

4 Offset Compensation Methods

All offset compensation methods described in this section are implemented in the Null_Offset program. The reader is invited to review the program code and to evaluate the method's effectiveness.

4.2 Device Offset Calibration Commands

The SFR register ADCON1 controls the device offset calibration command.

The advantage of this approach is its simplicity. It does not need any additional program memory; it uses only seven ADC samples, and it is fully automatic. This solution does not always work well, however, especially if you have low ENOB, high PGA gain, or are not using an input buffer.

In the case of system-offset calibration, the user should also provide a signal to the device, in addition to fixed and stabilized input voltage. This signal informs the program that system calibration can begin. The user should also keep in mind that during system-offset calibration, the input signal should have minimal noise.

4.3 Using Offset DAC

The SFR register ODAC controls the Offset DAC.

The offset DAC is specifically created to compensate an offset or shift ADC results. Its limitation of usage is range and resolution. The Offset DAC has 254 active codes in a range from $-V_{REF}/2$ to $+V_{REF}/2$ in bipolar ADC mode, which means one DAC LSB is equivalent to 1968.5ppm (or 0.2%) of ADC input range, or 9.8mV with $V_{REF}=2.5$ V. The ADC scale lies from 0 to V_{REF} in unipolar mode, but the offset DAC in this case ranges from $-V_{REF}/2$ to $+V_{REF}/2$, which also narrows the DAC use. Another complication is that an Offset DAC code is sign magnitude binary, and the user must make some code conversion before loading data to the ODAC register. (See Table 3.)

ODAC_error from the above formula mainly comes from two sources: Offset DAC gain error (typically, $\pm 0.5\%$ of FSR), which shows how precisely the DAC end codes match to $\pm V_{REF}/2$; and Offset DAC linearity, which is shown in Figure 12. If the user program works with Offset DAC, it should zero the DAC code before executing the offset calibrating command; otherwise, calibration will compensate the offset DAC action.

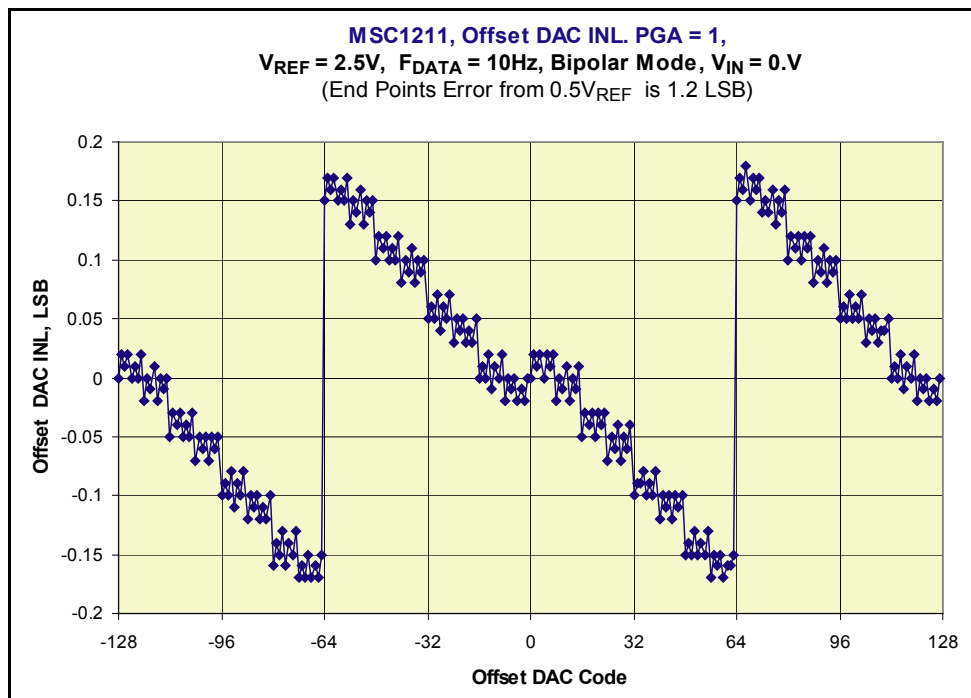


Figure 12. Offset DAC INL.

Table 3. ODAC codes (in bipolar mode)

ODAC Code	ODAC Voltage Unipolar/Bipolar Mode
1111 1111	+0.5 * V _{REF}
1000 0000	0V
0000 0000	0V
0111 1111	-0.5 * V _{REF}

4.4 Offset Compensation by Swapping Inputs

As mentioned before, swapping inputs can change the offset sign, but the absolute value stays approximately the same. This feature can be used to calculate offset out from the steady signal. For example: The program collects 10 samples when AIN0 is the negative ADC input and AIN1 is the positive ADC input. The program then swaps the inputs vice-versa, discarding three to five samples to give time for the digital filter to settle, then collects 10 new samples and inverts them. The program now calculates an average from the 20 collected samples. This method can be explained by the following formulas:

$$AD_P = V_{IN1} - V_{IN2} + \text{offset}$$

$$AD_N = V_{IN2} - V_{IN1} + \text{offset}$$

$$AD \text{ Total} = (AD_P - AD_N)/2 = (V_{IN1} - V_{IN2} + \text{offset} - V_{IN2} + V_{IN1} - \text{offset})/2 = V_{IN1} - V_{IN2}$$

This method decreases the offset, but cannot guarantee the full offset compensation.

4.5 Offset Subtraction from ADC Result

During the software offset calibration cycle, the program calculates the input average value with necessary precision and stores it. When data collection starts, the program subtracts the average value from each coming sample. This method is the most universally accepted and the only one that does not narrow the ADC input range. The disadvantage to this approach is that each time the ADC data is received, we need at least 12 assembler commands to execute a 24-bit subtraction process, or five assembler commands if the summation register is used for subtraction.

4.6 Placing Necessary Offset Value in OC Register

How does the method work? After gain calibration is complete, the input signal V_{INy} , which also can be 0 volts, is applied to the ADC input, and the conversion result is acquired. Now the question arises, What value should we put in the offset calibration register (OCR) so that the AD conversion result with V_{INy} input signal would be zero?

If $ADCRES_y$ is the A/D conversion result, when some V_{INy} input signal is applied, then we can convert the formula (A/D) to this equation:

$$0 = [(V_{INy}/V_{REF}) * 0.75 * BG * DEC^3 - (OC + ADCRES_y * N/GC)] * GC / N \quad (3)$$

If we identify the value in the last parentheses as OC_y , then

$$OC_y = OC + ADCRES_y * N / GC \quad (4)$$

and Equation (3) becomes

$$0 = [(V_{INy}/V_{REF}) * 0.75 * BG * DEC^3 - OC_y] * GC / N \quad (5)$$

This means having V_{INy} as input signal and OC_y value in offset calibration registers will create zero ADC result. If $ADCRES_y$ is an offset value that we want to compensate or a desired input signal shift, then after gain calibration, using equation (4), we can calculate the new value for offset register. There is no need in this case to apply the self-offset calibration command, and therefore the OC value in equation (4) will be equal to zero. The advantage of this method is that we perform the calculation only once during the calibration; subsequently, all subtractions are done by device hardware. Multiplication of the $ADCRES_y$ value by $N = 2^{22}$ demands a 48-bit word width. This can be done using a C-language floating-point library, which requires some additional program memory.

4.7 Calculation During Program Design State

If the external input signal has some constant shift that is known at the program design time, you may want to put some value in the offset register that would compensate this shift.

Analyzing the A/D formula, you can notice that it is also possible to get a zero ADC result when

$$[(V_{IN}/V_{REF}) * 0.75 * BG * DEC^3 - OC] = 0 \quad (6)$$

In this case, the necessary value for the offset register to compensate V_{INy} input signal will be

$$OC_y = OC + (V_{INy}/V_{REF}) * 0.75 * BG * DEC^3 \quad (7)$$

Because the decimation ratio and V_{REF} are not usually changed in the program, we can introduce a constant $K = 0.75 * BG * DEC^3$, which can also be calculated during the program design. Now the equation is:

$$OC_y = OC + V_{INy}/V_{REF} * K \quad (8)$$

OC is a value, loaded into the offset calibrating registers by self-offset calibration command. If, after reset, the program does not perform the offset calibration command, the OC value equals zero.

For example, in the Null_Offset program the default decimation is equal to 1562, then BG from Table 2 is 2-9, and

$$K = 0.75 * (15623) * 2^{-9} = 5582572.75 \quad (9)$$

When this formula is used to compensate offset measured by the microprocessor program itself, V_{INy} is an A/D conversion result during an offset measurement, and V_{REF} value is equivalent to 2^{23} in bipolar mode case, and to 2^{24} in unipolar mode.

When this formula is used to compensate an input voltage during the program design, please note that the V_{INy} in this case is a value created by formula (IN). This means a higher possible error for a greater PGA gain. For PGA gain = 1, internal experiments have shown an offset compensation precision greater than 0.1%.

5 The Null_Offset Program

The program Null_Offset is designed to give the user the ability to try different kinds of offset compensation methods in order to determine one that best suits the specific application.

The program is written in C language. It communicates with the serial terminal through device serial port 0 and, with default F_{CLK} , can support communication speeds of 4800 BPS or higher.

By default, the program has settings which best fit the evaluation board MSC1210-DAQ-EVM:

$AV_{DD} = 5.V$, V_{REF} internal = 2.5V, $F_{CLK} = 1.84MHz$, ADC Buffer OFF, bipolar mode, digital filter Sinc3, PGA gain = 1, Decimation value = 1562. This configuration produces a data rate of $\pm 9.2Hz$, ADC positive input = AIN0, ADC negative input = AINCOM. The user can change each of these settings. The size of the program is 4256 bytes, which allows use of the device level Y3 or higher.

After downloading the program to the device, the user should apply some steady input voltage to the inputs AIN0 - AIN7. The recommended starting point is an input differential voltage 1V and an input common voltage 1.5V.

On reset, the Null_Offset program sets the ADC and waits for the CR character (a carriage return, or the "Enter" key) from the serial terminal. When the CR is received, the program performs a self-gain calibration and averages three samples of 16 ADC counts, using the summation register. Each averaging result is submitted to the terminal. After this is complete, the program asks the user to enter one of the following numbers, which correspond to the next offset compensation method:

- 0 No offset compensation
- 1 Offset self calibration command
- 2 Offset system calibration command
- 3 Offset compensation using Offset DAC (Input signal less than $0.5 * V_{REF}$)
- 4 Offset compensation by input swapping
- 5 Offset compensation by subtraction from the ADC result
- 6 Offset compensation using formula (2)
- 7 Offset compensation using formula (6)

When the user enters the compensation method number, the program performs the compensation and goes into an infinite ADC loop, sending out the decimal results of averaging 16 samples. Remember that the greatest number in bipolar mode is 8388607. When using method 1 or 4 there will be a slight change in results, caused by offset remove. In all other cases, the ideal (correct) result after calibration should be at or near zero.

6 Recommendations

Choose the offset compensation method which best fits your task.

Use ADC input buffer if possible.

Use ADC neighbor pins.

The higher the PGA gain, the more stabilized power supplies must be.

The higher the PGA gain, the lower signal common voltage change is allowed.

For high-precision measurements, track the device temperature, and recalibrate ADC if necessary.

Conclusion

ADC offset error, besides ENOB, INL and input range, plays an important role in the MSC12xx performance. Underestimating this parameter can easily affect the expected results. The acceptable level of offset error should be set during the design phase. Once selected, the most suitable method of offset compensation needs to be identified. The Null_Offset program can be an effective tool in this process, and can be used as the source C code example.

References

1. *MSC1210 Product Data Sheet* <http://focus.ti.com/lit/ds/symlink/msc1210y2.pdf>
2. *MSC1211 Product Data Sheet* <http://focus.ti.com/lit/ds/symlink/msc1211y2.pdf>
3. *MSC1212 Product Data Sheet* <http://focus.ti.com/lit/ds/symlink/msc1212y2.pdf>
4. Application Note: *Calibration Routines and Register Value Generation for the ADS121x Series* (SBAA099)
5. Application Note: *ADC Gain Calibration – Extending the ADC Input Range in MSC12xx Devices* (SBAA107)
6. Application Note: *Input Currents for High-Resolution ADCs* (SBAA090)
7. *MSC12xx: Null_Offset program* (executable)

To obtain a copy of the referenced documents, visit the Texas Instruments website at www.ti.com.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		Applications	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265