Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

www.infineon.com

AN68403

PSoC® 3 and PSoC 5LP Analog Signal Chain Calibration

Author: Gautam Das G and Praveen Sekar

Associated Part Family: CY8C38xx, CY8C58xx

Associated Code Examples: Yes

Related Application Notes: AN84783

AN68403 explains how to calibrate an analog signal chain by using a calibrated Delta Sigma ADC and an on-chip EEPROM that are available in PSoC® 3 and PSoC 5LP. An example of a programmable gain amplifier as part of the analog signal chain is also described. AN68403 shows how gain and offset errors can be eliminated in the entire signal chain.

Contents

1 Introduction 1
2 Analog Signal Chain Errors 1
3 Calibration 3
3.1 ADC Calibration 3
3.2 Calibrating the Signal Chain 4
3.3 Routine to Write into GCOR and OCOR from EEPROM 11
4 Calibrating ADC Offset and Gain Using ADC Component APIs 12
5 Calibrating Offset and Gain when Using SAR ADC in PSoC 5LP 12
6 Summary 13
7 Related content 13
Document History 14
Worldwide Sales and Design Support 15

1 Introduction

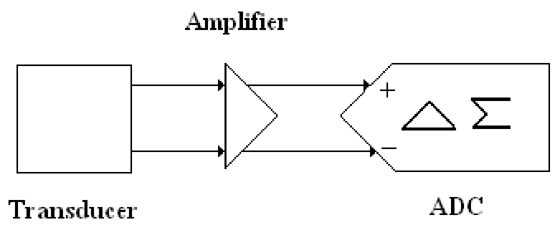
PSoC® 3 and PSoC 5LP have a 20-bit Delta Sigma Analog-to-Digital Converter (ADC). A typical analog signal chain consists of a sensor whose weak analog signal is amplified and is fed to an ADC, which converts it to a digital value. The amplifier that is used can be a programmable gain amplifier (PGA) or a transimpedance amplifier (TIA).

An amplifier block has inherent errors: mostly gain and offset errors. Because these errors propagate through the signal chain, the value obtained from the ADC deviates from the actual value. For an accurate measurement, calibration of the entire signal chain is required.

2 Analog Signal Chain Errors

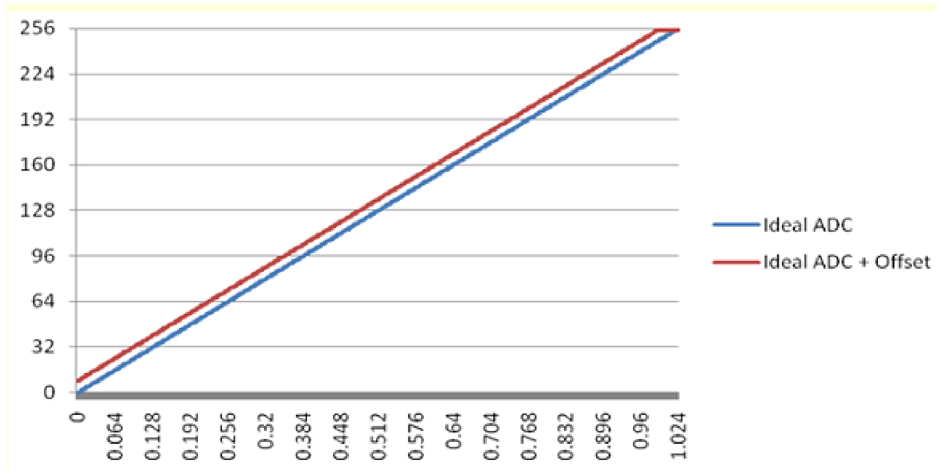
Figure 1 shows a simple analog signal chain that consists of a transducer with output in the form of analog voltage. This analog voltage is passed through an amplifier and then fed to an ADC.

Figure 1. Simple Analog Signal Chain



Offset error is the error that the offset voltage of the ADC creates. Figure 2 shows an ideal ADC transfer characteristic and one with offset. The characteristics shown are that of an 8-bit ADC that measures 0 to 1.024 V with an offset of 32 mV. The figure shows that the offset causes a fixed additive error in all measurements. Offset also causes a loss of ADC input voltage range. The output is at full capacity at 32 mV (offset) below full scale in the following plot that sets a maximum input of only 992 mV instead of 1.024 V. The output value for a zero input voltage defines the offset of the ADC. The ideal transfer curve passes through the 0 reading when the input voltage is 0.

Figure 2. Offset Error



The following equation gives the ideal ADC transfer function:

$$\text{Voltage} = \text{ADC Count} * \frac{\text{ADC Voltage Range}}{2^n}$$

Any multiplicative factor in this equation, as shown in the following equation, causes a gain error:

$$\text{Voltage} = \text{ADC Count} * \frac{\text{ADC Voltage Range}}{2^n} * k$$

Where *n* is the resolution of the ADC.

Figure 3 shows a plot of the above two equations with ADC counts along the Y-axis and input voltage along the X-axis. The graph shows an 8-bit ADC that measures from 0 to 1.024 V. The blue line represents the ideal transfer characteristic. The red line represents the characteristic with gain error (10 percent) (put k = 0.9 in the previous equation).

Figure 3. Gain Error

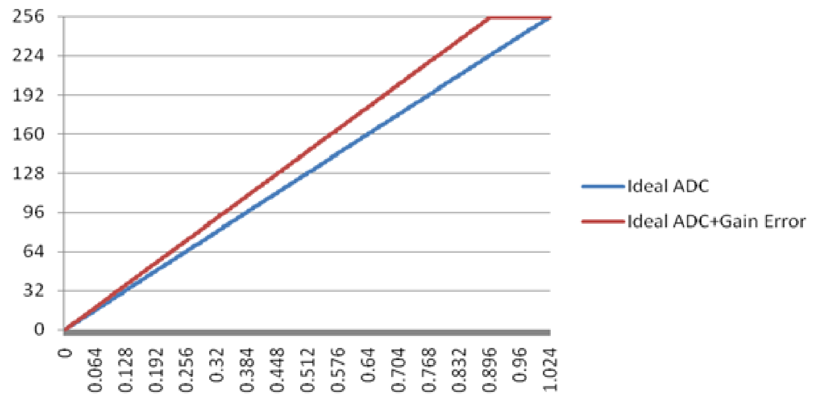
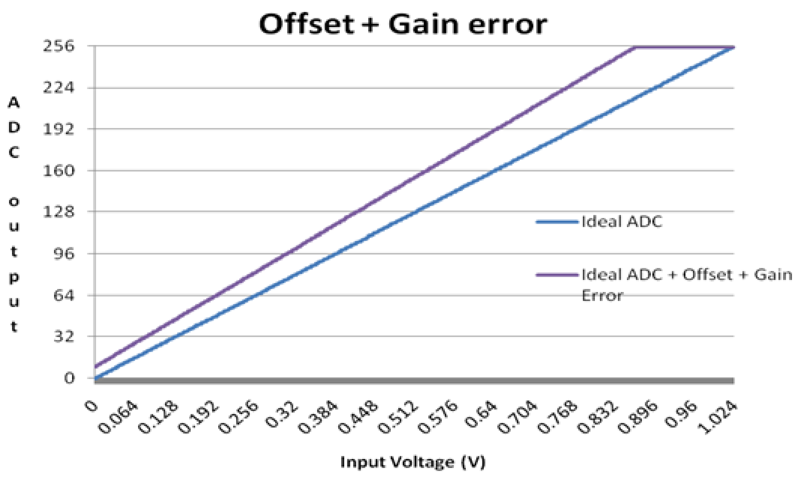


Figure 4 shows the effect of both the gain and offset error in a system. The blue line represents the ideal characteristic without gain and offset error, and the violet line represents the characteristic with gain and offset error.

Figure 4. Gain and Offset Error



3 Calibration

Calibration of an analog signal chain involves eliminating the gain and offset errors in the entire signal chain. Based on where and how it is performed, there can be different types of calibration such as the following:

- Manufacturing calibration
- User calibration
- Runtime calibration

In manufacturing calibration, the analog block under consideration is calibrated during the manufacturing process. This can be during IC manufacturing or assembly manufacturing. For example, the ADC in PSoC 3 is calibrated during IC manufacturing. However, a multimeter is calibrated as an assembly in the multimeter manufacturing plant.

In the user calibration method, the user calibrates the analog block used in the chain. As an example, some cameras have a mode to calibrate the level sensor. The user who initiates this mode does not require any standard except for a level surface. Another user calibration includes the periodic calibration of test equipment.

In the runtime calibration method, the analog block is calibrated in runtime for voltage offsets and system gain errors.

3.1 ADC Calibration

The Delta Sigma ADC available in PSoC has 20 input ranges that require calibration. This includes ranges Vref*2, Vref, Vref/2, Vref/4, Vref/8, and Vref/16 in differential mode; and Vss to Vref, Vref*2, Vdd and Vref*6 in the single-ended mode for 8-15 bits and 16-20 bits resulting in 20 input ranges. Because the calibration memory has room for eight ranges, the range that is most likely to be used has been calibrated. You can calibrate the remaining non-calibrated ranges using one of the calibrated ranges.

Table 1 shows the eight ranges that have been factory-calibrated.

Table 1. Calibrated ADC Ranges

	Resolution	Range
1	16-20 bits	+/- Vref (Differential)
2	16-20 bits	+/- Vref/2 (Differential)
3	16-20 bits	+/- Vref/4 (Differential)
4	16-20 bits	+/- Vref/16 (Differential)

	Resolution	Range
5	8-15 bits	+/- Vref (Differential)
6	8-15 bits	+/- Vref/2 (Differential)
7	8-15 bits	+/- Vref/4 (Differential)
8	8-15 bits	+/- Vref/16 (Differential)

The ADC calibration is done to correct any gain error that may be caused by process variations. The input gain is a function of the ADC input capacitor ratio. Slight process variations can cause these capacitors to vary in size and, therefore, affect the ADC input gain. The front-end ADC buffer is set to a gain of 1 during the calibration process. If the front-end buffer's gain is chosen to be any value other than 1, the factory calibration values no longer hold true.

The Delta Sigma ADC in PSoC has a post-processing block that can multiply the ADC result by a value between 0 and 2, with 16 bits of resolution. The registers, GCOR(LSB) and GCORH(MSB), hold the correction value and can be written during runtime to provide a gain correction factor between 0 and 2.

Table 2 shows the format of the GCORH and GCOR registers. Each bit is weighted between 1 and 1/ 32768, similar to an unsigned number, but with fractional bit weights. The Delta Sigma ADC Component API in PSoC Creator provides functions for writing the GCOR registers.

ADC_SetGCOR (float gainAdjust): This function calculates a new GCOR (ADC Gain) value and writes it into the GCOR registers. The GCOR value is a 16-bit value that represents a gain of 0 to 2. The ADC result is multiplied by this value before it is placed in the ADC output registers.

When executing the function, the old GCOR value is multiplied by gainAdjust input and reloaded into the GCOR register. The GCOR value is normalized based on the GVAL register.

The value, calculated by this API, is also stored into the RAM for each active configuration and used by SelectConfiguration() API to initialize the GCOR register.

Table 2. GCOR Registers

GCORH	15	14	13	12	11	10	9	8
	1	1/2	1/4	1/8	1/16	1/32	1/64	1/128

GCOR	7	6	5	4	3	2	1	0
	1/256	1/512	1/1024	1/2048	1/4096	1/8192	1/16384	1/32768

The OCOR registers are used to provide offset correction in an ADC. A 24-bit register consisting of 3 bytes, OCOR (LSB), OCORM, and OCORH (MSB), holds the correction value and can be written during runtime to provide offset correction. In the single-ended 0-to-2 Vref range, this value has an offset of about half the full-scale range for that resolution.

3.2 Calibrating the Signal Chain

The following is a generic procedure used to calibrate the signal chain:

1. Measure a stable voltage from the PSoC internal voltage DAC (VDAC) with one of the calibrated ranges. This is the "VDAC_Direct" value.
2. Measure the offset voltage of the system by grounding the input terminals. This is the "Sys_Offset" value.
3. Pass the same voltage from VDAC through the signal chain and measure the value. The reading obtained is the "SigChn_measured" value.
4. Offset-calibrate this reading is offset by subtracting the offset from it. The offset-calibrated reading is the "SigChn_offset_corrected" value.

   From the previous discussion, it can be written as

   SigChn_offset_corrected = SigChn_measured - Sys_Offset **Equation 1**

5. The observed gain (Observed_Gain) of the system can be calculated by dividing the offset-calibrated reading after passing through the signal chain by the original reading of the VDAC.

   Therefore,

   Observed_Gain = SigChn_offset_corrected / VDAC_Direct **Equation 2**

6. The ratio of the ideal gain to observed gain of the signal chain is computed. Call the ideal gain value "Ideal_Gain".

   Therefore, **Ratio = Ideal_Gain / Observed_Gain**

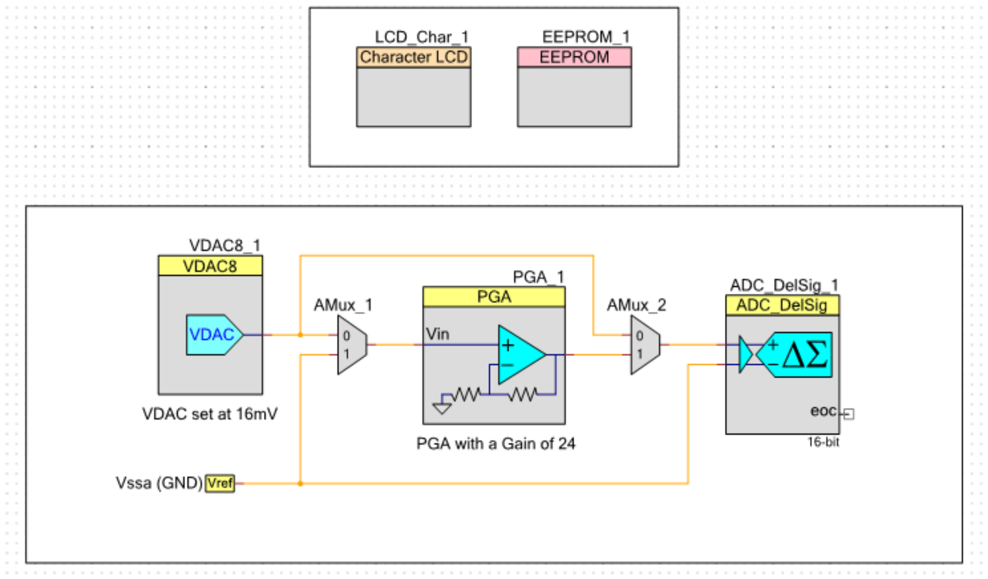
The ratio thus obtained is stored in the EEPROM to complete the process of calibration.

When the signal chain under consideration is used, the value stored in the EEPROM is written to the ADC gain correction and offset corrections registers.

A PSoC Creator project that writes the gain ratio in the EEPROM is attached with this application note. The DAC used in this process need not be accurate, but it should be stable with minimum drift. The calculated ratio is of interest, not the actual value of voltage itself.

A routine is provided in the Routine to Write into GCOR and OCOR from EEPROM section. When called in the target project, it writes the gain correction values obtained from the EEPROM to ADC gain correction registers and the offset correction values to the OCOR registers. Figure 5 shows the top design of the project.

Figure 5. Top Design

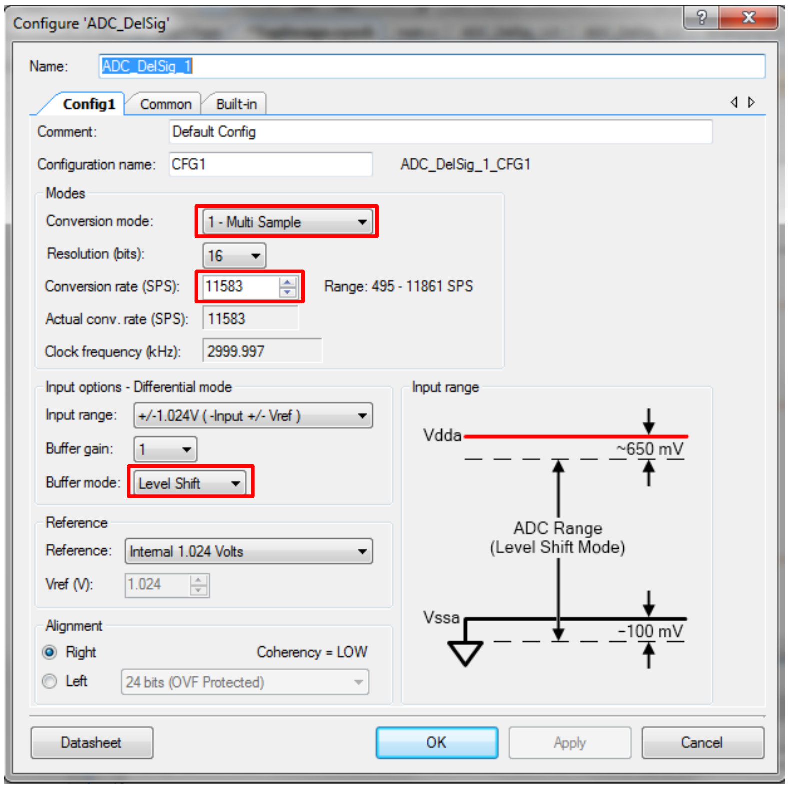


The configuration of individual components used in the project is described in the next section.

3.2.1 ADC Configuration

The configuration tab of the ADC is as shown in Figure 6. The parameters that are changed from the component defaults are highlighted in Figure 6.

Figure 6. Delta Sigma ADC



Configure 'ADC_DelSig'

Name: ADC_DelSig_1

Config1 | Common | Built-in

Comment: Default Config

Configuration name: CFG1 ADC_DelSig_1_CFG1

Modes

Conversion mode: 1 - Multi Sample

Resolution (bits): 16

Conversion rate (SPS): 11583 Range: 495 - 11861 SPS

Actual conv. rate (SPS): 11583

Clock frequency (kHz): 2999.997

Input options - Differential mode

Input range: +/-1.024V (-Input +/- Vref)

Buffer gain: 1

Buffer mode: Level Shift

Reference

Reference: Internal 1.024 Volts

Vref (V): 1.024

Alignment

Right [✓] Coherency = LOW

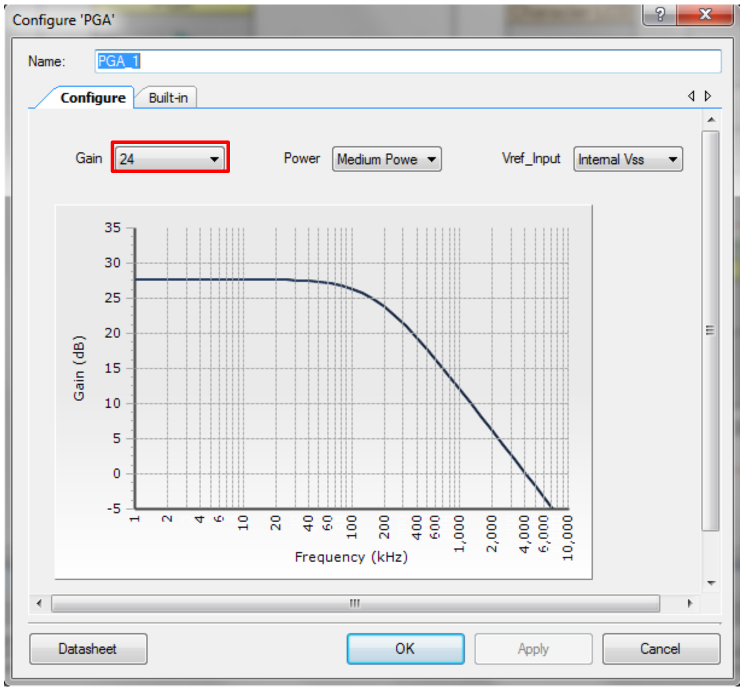
Left 24 bits (OVF Protected)

The resolution of the ADC is set to 16 bits, which is used in the differential input mode with an input range of ±Vref. The conversion rate is 11,583 samples per second and the conversion mode is set to Multi Sample mode. The Buffer mode is set to “Level Shift”. The level shift mode lets the ADC measure range down to Vssa. Because the offset measurement happens close to Vssa, the Buffer mode should be “Level Shift”.

3.2.2 Programmable Gain Amplifier (PGA)

The PGA used in the top design forms a part of the analog signal chain. The gain of the PGA can be written during runtime. In this case, the gain is set to 24. The Configuration tab is as shown in Figure 7. The gain parameter is highlighted.

Figure 7. Programmable Gain Amplifier


Configure 'PGA'

Name: PGA_1

Gain 24

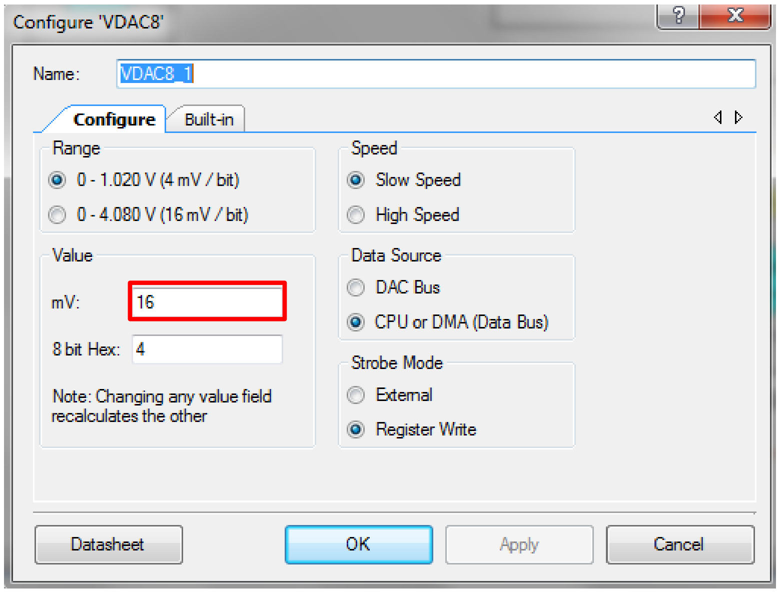
Power Medium Powe

Vref_Input Internal Vss

3.2.3 Voltage DAC (VDAC)

The VDAC Component used has been configured to output 16 mV. The Configuration tab is as shown in Figure 8. The highlighted parameter shows the setting of the VDAC output to 16 mV.

Figure 8. Voltage Digital-to-Analog Converter



Configure 'VDAC8'

Name: VDAC8_1

Range
- [✓ 0 - 1.020 V (4 mV / bit)]
- 0 - 4.080 V (16 mV / bit)

Speed
- [✓ Slow Speed]
- High Speed

Value

mV: 16

8 bit Hex: 4

Note: Changing any value field recalculates the other

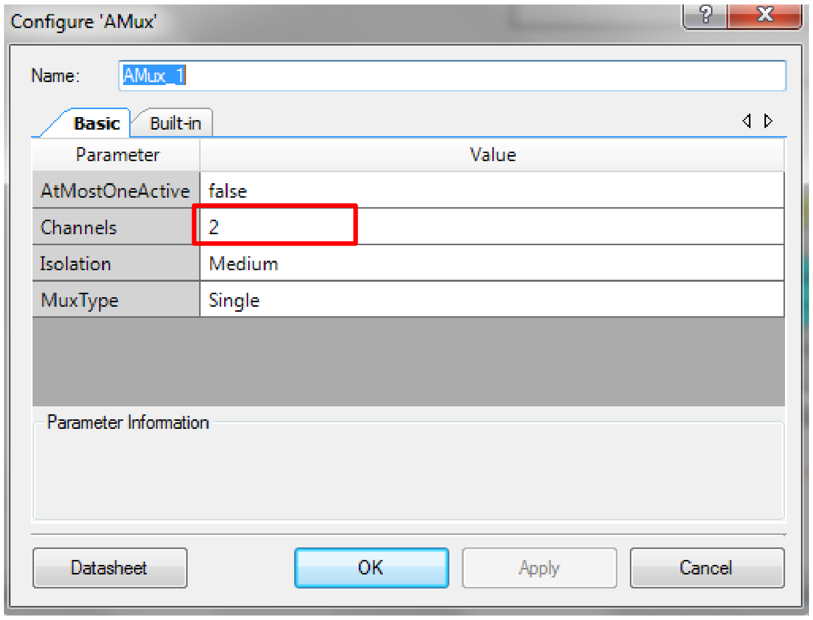
Data Source
- DAC Bus
- [✓ CPU or DMA (Data Bus)]

Strobe Mode
- External
- [✓ Register Write]

3.2.4 Analog Multiplexer (AMux)

Two software analog multiplexer (AMUX) components are used with two input channel and Single MuxType. This is used to multiplex the analog signals to the PGA and ADC. The Configuration tab is as shown in Figure 9.

Figure 9. Analog Multiplexer (AMux)



Configure 'AMux'

Name: AMux_1

Parameter	Value
AtMostOneActive	false
Channels	2
Isolation	Medium
MuxType	Single

Parameter Information

On-chip EEPROM is used to store the GCOR and OCOR values computed in the project.

An LCD Component is used to display the GCOR and OCOR values computed.

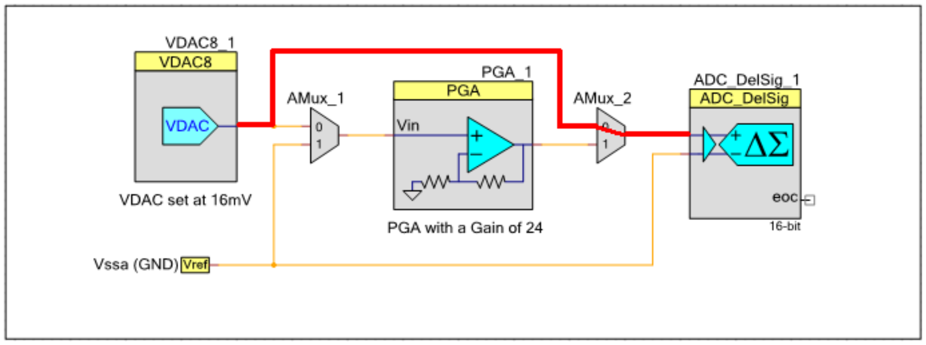
3.2.5 Calibrating the Analog Signal Chain

When any analog block such as a PGA is cascaded with an ADC, the gain and offset errors of that block affect the entire signal chain. A calibrated ADC is used to calibrate the entire signal chain and compensate for the errors introduced by the analog block. The following procedure is used to calibrate the analog signal chain. See Figure 5 for the complete schematic of the project.

Step 1

AMux_2 channel 0 is selected. This connects the VDAC output to the ADC. PGA is not used in the signal path in this configuration. This gives a direct reading of VDAC voltage. The samples are averaged. As discussed previously, consider this to be value VDAC_Direct. Figure 10 shows the signal flow for this step where the red line shows the path taken.

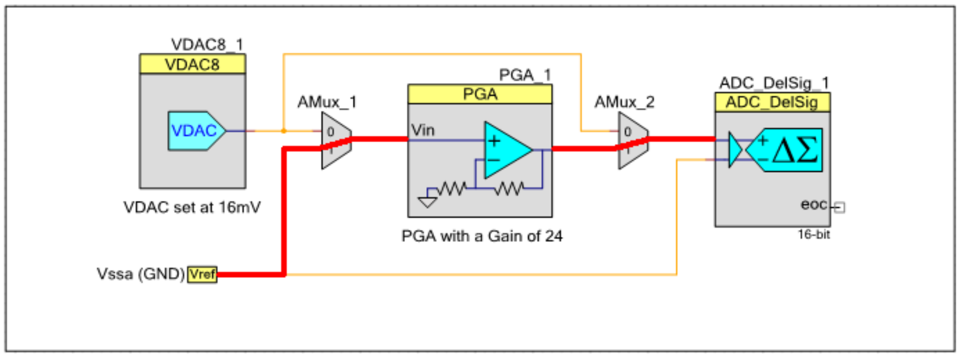
Figure 10. VDAC Direct Measurement



Step 2

The AMUX1 and AMUX 2 connect to their respective channel 1. This sets up the signal chain to measure the ground potential at the input of the PGA. This reading corresponds to the offset error of the PGA. Consider this to be the value Sys_Offset. The red line in Figure 11 shows the path taken by the analog signal.

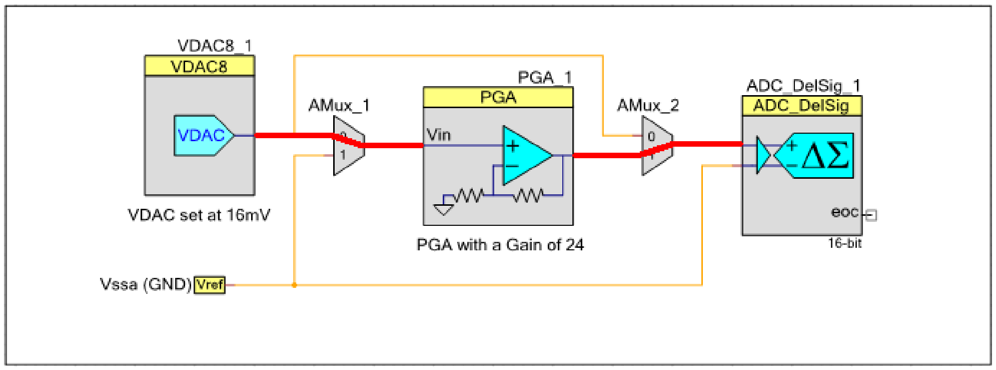
Figure 11. Offset Error Measurement



Step 3

AMux_1 channel 0 and AMux_2 channel 1 are selected, which pass the VDAC output through the PGA. This gives the PGA output that has gain as well as offset error. Consider this measured value as value “SigChn_measured”. The red line in Figure 12 shows the path taken by the analog signal.

Figure 12. VDAC Output Passes Through PGA



Step 4

Offset error is removed from this reading by subtracting the value obtained in Step 2 from that of Step 3. This measurement corresponds to a value free from offset error. Consider this as “SigChn_offset_corrected”. From the definition, the value of SigChn_offset_corrected can be computed as follows:

SigChn_offset_corrected = SigChn_measured– Sys_Offset

Step 5

The gain of the PGA is obtained by dividing the offset-free measurement SigChn_offset_corrected by VDAC_Direct obtained in Step 1. It is mathematically written as:

Observed_Gain = SigChn_offset_corrected / VDAC_Direct

Step 6

The ideal gain of the PGA, “I” (which in this case is 24), is divided by the actual gain obtained, “G”. This is the ratio that must be written into the EEPROM.

Ratio = Ideal_Gain / Observed_Gain = 24 / Observed_Gain

Step 7

PSoC 3 and PSoC 5LP have a switched capacitance (SC)/ continuous time (CT) block, which is a general-purpose block, constructed on a rail-to-rail amplifier with arrays of switches, capacitors, and resistors. PGA is a CT opamp with selectable taps for input and feedback resistors. There are four SC/CT blocks available in PSoC 3 and PSoC 5LP. Because the gain and offset errors of the PGA differ depending upon the SC block used, it is necessary to force the fixed SC block for a particular design. This is done in the directives tab of the *.cydwr* of the project.

Figure 13 shows the settings used to force the SC3 block.

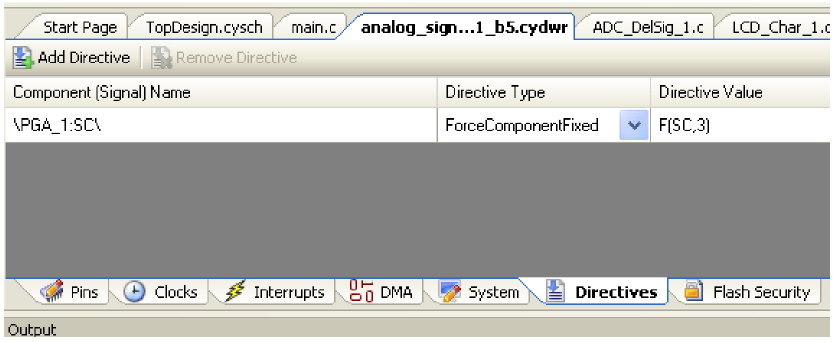
In the “Component (Signal) Name” tab, the name of the Component is written, which in this case is PGA_1 followed by SC, which is separated by a colon. This is written between backslashes.

In the “Directive Type”, ForceComponentFixed is chosen to force one specific SC block among the available four blocks (0, 1, 2, and 3) to implement the PGA.

The “Directive Value” tab is used to select the required SC block to be used for the given Component. In this case, SC3 is chosen to implement PGA_1; therefore, the Directive Value is F (SC, 3).

The placement of PGA_1 can be confirmed by verifying the report file (*.rpt*) in the project.

Figure 13. Forcing the SC Block for PGA



Component (Signal) Name	Directive Type	Directive Value
\PGA_1:SC\	ForceComponentFixed	F(SC,3)

Step 8

The gain ratio and offset error value thus obtained are written into the on-chip EEPROM.

Step 9

In the final target project where the particular signal chain is to be used, the gain ratio and the offset error values are read from the EEPROM.

Step 10

The gain ratio read from the EEPROM is updated to the GCOR register using ADC_setGCOR(). The offset error value read is added to the current value of OCOR and the result is written back to the OCOR register. GCOR is enabled.

3.3 Routine to Write into GCOR and OCOR from EEPROM

Assume that “gcor_new” is the gain value read from the EEPROM. Assume that “ocor_new” is the offset value read from the EEPROM..

The existing OCOR value is read from the OCOR registers and stored in “ocor_old”. The new OCOR value is calculated by adding ocor_new to ocor_old. This result is written back into the OCOR register.

Updating the GCOR register is achieved using ADC_SetGCOR().

The code snippet below implements the updating of the OCOR and GCOR registers from the EEPROM.

(Assume that the name of ADC is ADC_DelSig_1)

```
/* Read the existing OCOR register to ocor_old */
ocor_old = (int32) (ADC_DelSig_1_DEC_OCOR_REG) + ((int32)
(ADC_DelSig_1_DEC_OCORM_REG) << 8) + ((int32) (ADC_DelSig_1_DEC_OCORH_REG) <<16);
/* Add the offset correction read from EEPROM to the existing OCOR value */
ocor = ocor_old + ocor_new;
/* Write the updated OCOR value back to registers */
ADC_DelSig_1_DEC_OCOR_REG = (int8)(ocor);
ADC_DelSig_1_Dec_OCORM_REG= (int8)(ocor>>8);
ADC_DelSig_1_DEC_OCORH_REG= (int8)(ocor>>16);

/* Update the GCOR register with the GCOR adjustment read from EEPROM */
ADC_DelSig_1_SetGCOR(gcor_new);
```

4 Calibrating ADC Offset and Gain Using ADC Component APIs

The method of calibration explained in the preceding sections of this application note uses the GCOR and OCOR registers in the ADC hardware. This means that once the calibration routines have determined the GCOR and OCOR values and populated them in the respective registers, error correction is achieved as a part of the ADC hardware; no CPU bandwidth is consumed for this operation.

You can also implement error correction in firmware by using the following API functions in the ADC Component.

- ADC_CountsTo_mVolts
- ADC_CountsTo_uVolts
- ADC_CountsTo_Volts
- ADC_SetOffset
- ADC_SetGain

The ADC_CountsTo_mVolts, ADC_CountsTo_uVolts and ADC_CountsTo_Volts functions take raw ADC counts as input parameters, convert them, and return them in a voltage scale. The voltage is calculated based on the gain and offset configured by ADC_CountsPerVolt and ADC_offset parameters respectively.

The ADC_countsPerVolt variable is used to calibrate the gain. Initially, this variable is calculated for the default ADC configuration. The calculated value depends on the resolution, input range, and voltage reference. Applications can modify it using the ADC_SetGain() function. It affects only the ADC_CountsTo_Volts(), ADC_CountsTo_mVolts(), and ADC_CountsTo_uVolts() functions by supplying the correct conversion between ADC counts and the applied input voltage.

The ADC_offset variable is used to calibrate the offset. Initially, this variable is set to zero. Applications can modify it using the ADC_SetOffset() function. It affects only the ADC_CountsTo_Volts(), ADC_CountsTo_mVolts(), and ADC_CountsTo_uVolts() functions by subtracting the given offset.

ADC_SetOffset (int32 Offset): This function sets the ADC offset, which is then used by the functions ADC_CountsTo_uVolts(), ADC_CountsTo_mVolts(), and ADC_CountsTo_Volts() to subtract the offset from the ADC reading before calculating the measured voltage.

ADC_SetGain (int32 adcGain): This API sets the ADC gain in counts per volt for the voltage conversion functions ADC_CountsTo_uVolts(), ADC_CountsTo_mVolts(), and ADC_CountsTo_Volts().This value is set by default based on the reference and input range settings in the ADC component.

You can use the same calibration procedure detailed in the Calibrating the Analog Signal Chain section. The gain parameter would need to be calculated as counts to voltage ratio, rather than actual gain to observed gain ratio. In this case, the adcGain passed as part of the ADC_SetGain function would be

adcGain = **SigChn_offset_corrected / Actual VDAC voltage**

5 Calibrating Offset and Gain when Using SAR ADC in PSoC 5LP

The techniques explained in the Calibration section can be used for calibrating a signal chain with the SAR ADC too. The difference would be in the use of the GCOR and OCOR registers, which are not present in the SAR ADC in PSoC 5LP. So, the design would have to implement the gain and offset correction in code by using the Component APIs as explained in the Calibrating ADC Offset and Gain Using ADC Component APIs section.

The SAR ADC Component has API functions ADC_SetGain and ADC_SetOffset. ADC_SetOffset() can be used to set the offset correction after step 2 in Calibration. ADC_SetGain() can be used to set the counts per voltage value. For this, the offset corrected VDAC value from Step 4 in Calibration is divided by the actual VDAC voltage that was expected.

adcGain = **SigChn_offset_corrected (in counts) / Actual VDAC voltage (in volts)**

This adcGain value is set as the Gain value using ADC_SetGain(). The adcGain value is in counts per volts and sets up the scale factor for the ADC_CountsTo_mVolts, ADC_CountsTo_mVolts and ADC_CountsTo_mVolts API functions. During normal ADC sampling, the user will have to use these functions to convert the ADC result so that the gain and offset corrections are applied and the result is provided as a voltage.

6 Summary

The analog signal chain can be calibrated by using a VDAC, EEPROM, and a calibrated Delta Sigma ADC in PSoC 3 and PSoC 5LP.

7 Related content

Kits	
Kit	**Kit Description**
CY8CKIT-001	PSoC Development kit: Common development platform for all PSoC family devices.
CY8CKIT-030	Designed for analog performance. Enables you to evaluate, develop, and prototype high-precision analog, low-power, and low-voltage applications powered by PSoC 3.
CY8CKIT-050	Designed for analog performance. Enables you to evaluate, develop, and prototype high-precision analog, low-power, and low-voltage applications powered by PSoC 5LP
Application Note	
Document	**Document Name**
AN84783	Accurate Measurement Using PSoC® 3 and PSoC 5LP Delta-Sigma ADCs
AN65977	PSoC® 3 and PSoC 5LP - Creating an Interface to a TMP05/TMP06 Digital Temperature Sensor
AN66444	PSoC® 3 and PSoC 5LP Correlated Double Sampling to Reduce Offset, Drift, and Low Frequency Noise
AN75511	PSoC® 3 / PSoC 5LP - Temperature Measurement with a Thermocouple
AN70698	PSoC® 3 and PSoC 5LP – Temperature Measurement with an RTD
Component Datasheet	
Document	**Component Name**
ADC	Delta Sigma Analog to Digital Convertor
PGA	Programmable Gain Amplifier
VDAC	Voltage Digital to Analog Convertor
AMUX	Analog Multiplexer

Document History

Document Title: AN68403 – PSoC® 3 and PSoC 5LP Analog Signal Chain Calibration

Document Number: 001-68403

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	3205526	DASG	03/23/2011	New application note
*A	3441350	DASG	11/21/2011	Template update Updated Software Version to PSoC Creator™ 2.0 Updated snap-shots of ADC, PGA and VDAC Conversion rate of ADC is changed from 10,000 to 11,583.
*B	3564143	DASG	03/28/2012	Updated author's contact information according to the template. Added definition of Offset error and Gain error. The reference made to AN60263 for gain and offset error definition is removed. The project is updated with latest components.
*C	3642517	DASG	06/11/2012	Updated template to current CY standards. Updated associated project.
*D	3819305	DASG	11/22/2012	Updated for PSoC 5LP
*E	3889066	DASG	01/29/2013	Corrected headers and footers Sunset review
*F	5326880	QVS	07/05/2016	Screenshots updated to reflect PSoC Creator 3.3 changes Added section 6 for using the ADC component API for calibration. Added section 7 for SAR ADC support Updated associated projects and components to latest tool version Eliminated the use of floating point arithmetic in the project Commented the code for clearer demarcations on steps Updated template
*G	5688031	AESATMP8	04/19/2017	Updated logo and Copyright.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2011-2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.