NXP Semiconductors
Application Note

Document Number: AN5314
Rev. 1, 08/2016

ADC Calibration on Kinetis E+ Microcontrollers

1. Introduction

This document explains the basics of ADC calibration on KE+; the new generation of Kinetis E microcontrollers. The document also lists the steps to run the calibration from an application point of view on the KE+ microcontroller, which includes the KE15Z and KE18F series of microcontrollers.

Contents

2. ADC Calibration Overview

The 12-bit ADC on KE+ MCU is equipped with a calibration mechanism to provide high accuracy as specified in the data sheet. It is mandatory to calibrate the ADC after each system reset including power on reset. Failing to calibrate the ADC after each system reset can result in inaccuracy in the ADC conversion results. The calibration must be executed at least once after a power-on by the user to set the calibration start bit, and the user must check the calibration result as an option.

Compared to the previous ADC module, ADC on KE+ MCU removes the "CALF" bit, which indicates whether the calibration is successful. The user must check whether the calibration has failed by comparing the calibration result value with its limit range.

The main purpose of the ADC calibration is to generate the offset and gain compensation values. These values are automatically subtracted (offset) and scaled (gain) during the conversion sequence to compensate for linearity errors in the SAR's internal DAC output.

The calibration algorithm running inside the ADC module generates the offset correction value, the calibrated gain value, and other calibration values. The offset calibration value is automatically stored in the ADC offset correction register (OFS), the calibration gain value is stored in the ADC calibration gain register (G), and the other individual calibration values are automatically stored in each calibration registers, CLPx. The user must configure the ADC module correctly prior to calibration.

3. ADC Calibration Algorithm

3.1. SAR ADC Basics

In order to better understand the ADC calibration algorithm, it is necessary to know firstly how SAR ADC works. Taking a 4-bit ADC as a simple example to become familiar with the SAR ADC binary search algorithm, as shown in Figure 1. When the V_{REF} is 5 V, and analog input V_{IN} is 4.22 V ((27/32) *V_{REF} =4.22V), the 4-bit SAR ADC will implement 4 rounds of comparison to output the ADC code.

1. Set the MSB bit [3] as 1, 1000b. The DAC output is 2/4 VREF, which is less than the analog input, so the MSB bit [3] is 1
2. Set the bit [2] as 1, that is 1100b, so the DAC output is 3/4 VREF, VDAC < VIN, so the bit [2] is 1.
3. Set the bit [1] as 1, 1110b, the DAC output is 7/8 VREF, VDAC > VIN, so the bit [1] is 0.
4. Set the LSB bit [0] as 1, 1101b, the DAC output is 13/16 VREF, VDAC < VIN, so keep the LSB as 0.

This gives a final ADC conversion code of 1101b, which means the analog input equals 1101b/2^N (N=4). The analog input voltage is 13/16 $V_{REF,}$ while the real input is 27/32 V_{REF}. The difference between the ADC code digital representation of the input value and the actual input voltage is named as the Quantization Error, which is always +/- ½ LSB for the optimal ADC.

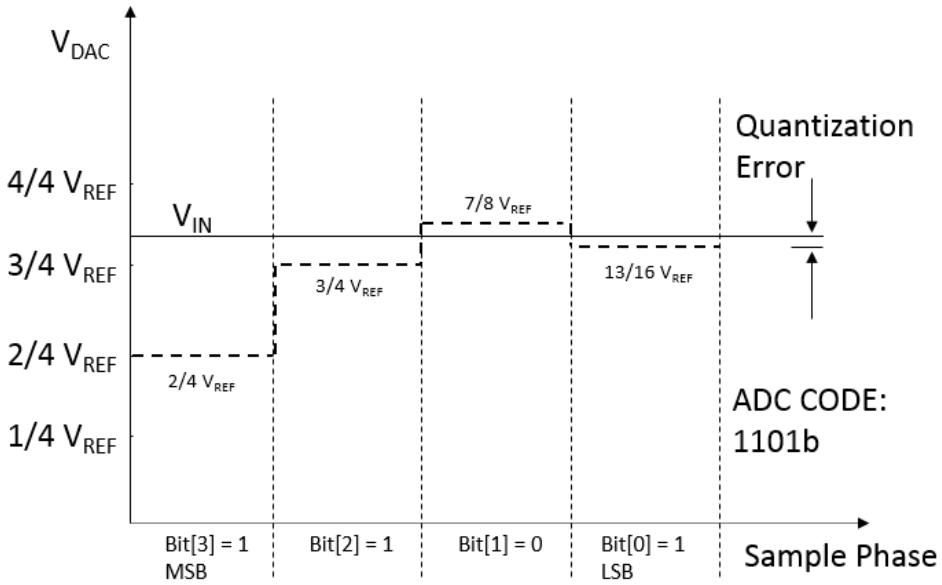

Figure 1. 4-bit SAR ADC example

3.2. ADB sub blocks

The SAR ADC consists of three sub-blocks:

1. SAR logic to implement binary search
2. Capacitive DAC to provide reference voltage
3. Comparator to determine whether the analog input V_{IN} is less than or greater than the V_{DAC}.

The ADC block diagram is shown in Figure 2. For ease of description, here we take the 4-bit ADC convert as an example.

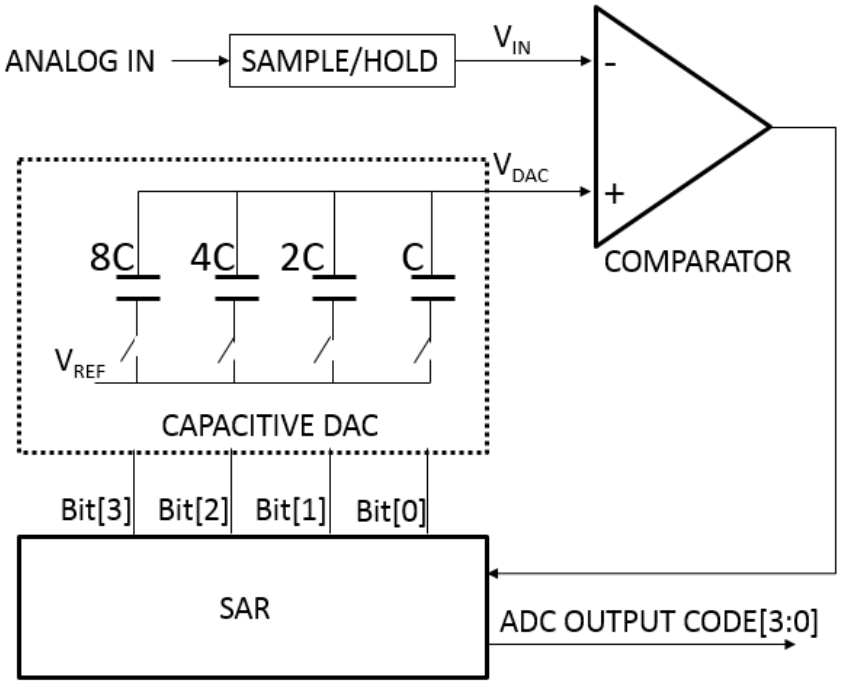

Figure 2. ADC sub-blocks

3.3. Offset error calibration

The offset error is defined as the deviation of the actual ADC transfer function from the ideal straight line at the zero input voltage. The offset error is also known as the zero-scale error, which indicates how well the actual ADC transfer function matches the ideal input at the zero point.

The offset error is mainly caused by a LSB capacitor size mismatch in the capacitive DAC, as shown in Figure 2. If the LSB capacitor [0] is undersized, for instance if the actual capacitor is smaller than pre-defined, then when comparing the bit [0] in the SAR binary search, $V_{IN} > V_{DAC}$ keeps bit [0] as 1. As a result, the undersized capacitor causes positive offset. This means that the first ADC code transition occurs at < -1/2LSB above zero, and vice versa, the oversized capacitor results in a negative offset error, which means the first ADC transition occurs > ½LSB. As the offset error is mainly caused by the LSB capacitor size mismatch, it can be calibrated by a small plus or minus offset correction.

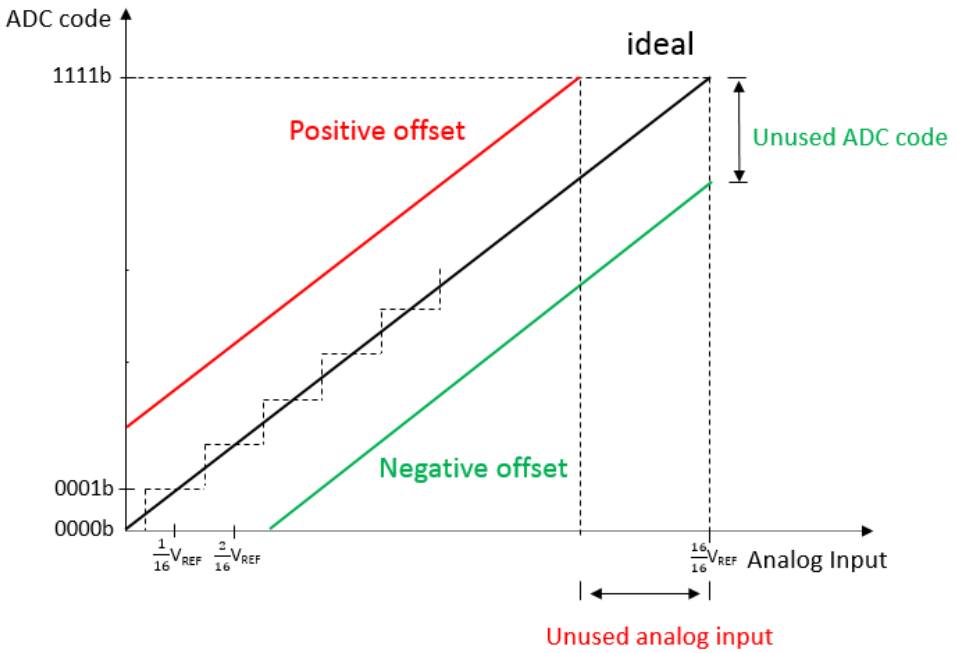

Figure 3. Offset error

3.4. Gain error calibration

Gain error is defined as the full-scale error minus the zero-scale offset error, which causes the actual transfer function to deviate from the ideal transfer function.

As shown in the 4-bit SAR ADC block diagram (Figure 5), the capacitive DAC error is accumulated. For example, if the bit [0] of the capacitive DAC is set, then the error is simply the error of bit [0] itself. However, if the bit [1] is set, the total error is bit [1] error plus bit [0] error. For each MSB the error is calculated as below, where Ex is the error found during the calibration for its corresponding MSB bit[x].

When,

- bit [0] is set: CLx0 = E0
- bit [1] is set: CLx1 = E1 + E0
- bit [2] is set: CLx2 = E2 + E1+ 2E0
- bit [3] is set: CLx3 = E3 + E2+ 2E1 + 4E0

From the error calculation above, we can see that the capacitor mismatch of the higher weighted capacitors (MSB) of the capacitive DAC mainly defines gain error, so the gain error can be calibrated out by adding additional charge in the undersized MSB capacitors of the capacitive DAC for the positive gain error, and moving charge in the oversized MSB capacitors.

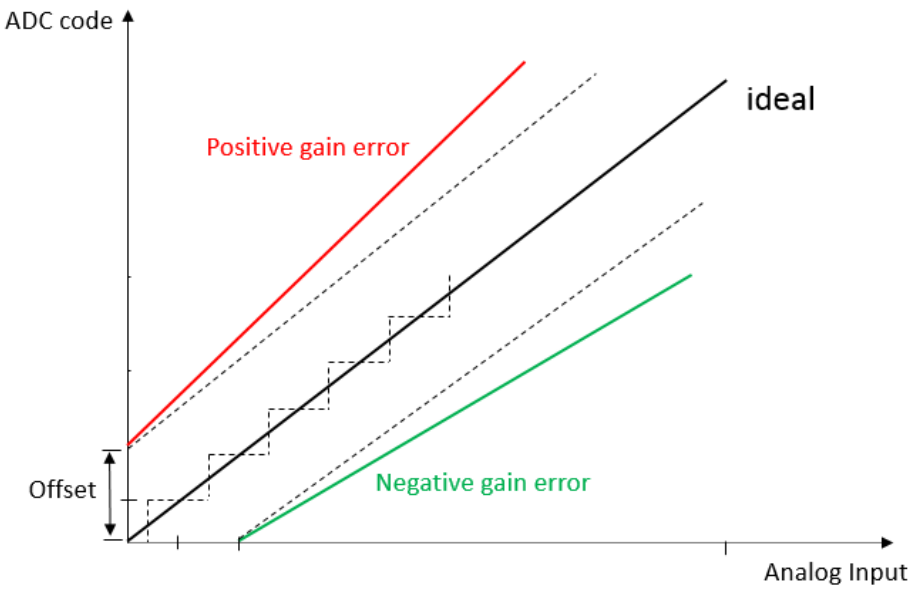

Figure 4. Gain error

3.5. ADC block diagram

Similar to the simple 4-bit SAR ADC module, the 12-bit SAR ADC also has the same algorithm, including the capacitive DAC, which consists of N capacitors with binary weighted values. Each capacitor should be exactly twice the value of next-smaller capacitor. The capacitive DAC provides an inherent track/hold function and uses the principle of charge redistribution to generate an analog output voltage to the comparator plus input.

The simplest capacitive DAC should include 12 capacitors, ranges from 4096C, 2048C, …, 8C, 4C, 2C, 1C, where C is the value of the LSB capacitor. However, the wide range of capacitor causes the realized analog size to be too large, so the N capacitors are divided into three capacitor arrays: MSB, INTERMEDIA, and LSB on the actual 12-bit ADC, as shown in Figure 5. The capacitors that represent the most significant bits (MSB) of the SAR (C14:C11) are connected directly to the inputs of the comparator. The next array of five capacitors (C10:C7) is connected to the top plate of the MSB array through an intentionally oversized scaling capacitor SC1. The final seven capacitors that make up the least significant bits(LSB) of the SAR (C6:C0) are correspondingly connected to the top plate of the INT Cap Array through another scaling capacitor SC2.

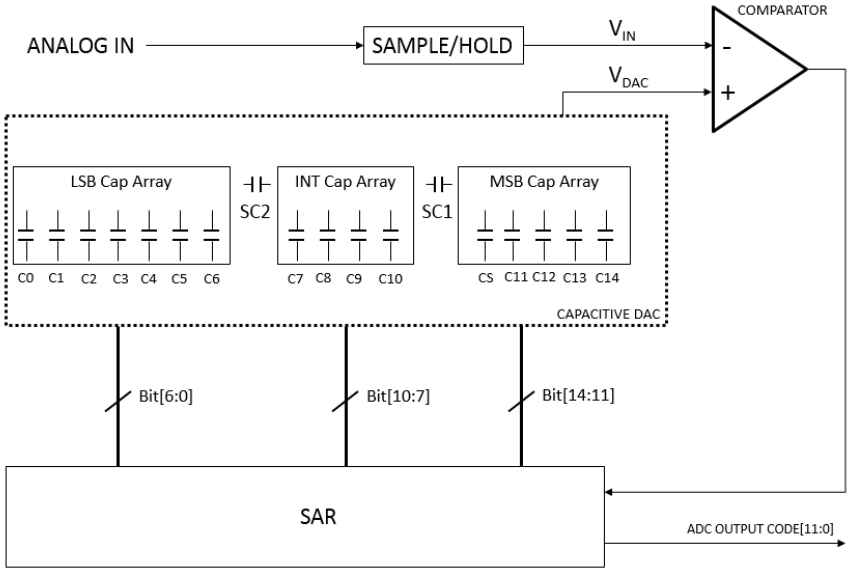

Figure 5. ADC block diagram

3.6. ADC calibration register function descriptions

Table 1. **ADC calibration register function descriptions**

Calibration Register Name	Function Descriptions
ADCx_OFS	ADC Offset Correction Register The offset calibration value is determined and uploaded by ADC self-calibration algorithm, and will be used for offset correction.
ADCx_G	Gain Correction Register The gain error correction for the overall conversion. It is a 12-bit gain adjustment factor in binary formats, this register value is determined and uploaded by the calibration algorithm.
ADCx_CLPS	CLPS Calibration Value Contains the capacitor Cs calibration value determined by the ADC self-calibration
ADCx_CLP3	CLP3 Calibration Value The capacitor C14 calibration value determined by the ADC self-calibration
ADCx_CLP2	CLP2 Calibration Value The capacitor C13 calibration value determined by the ADC self-calibration
ADCx_CLP1	CLP1 Calibration Value The capacitor C12 calibration value determined by the ADC self-calibration
ADCx_CLP0	CLP0 Calibration Value The capacitor C11 calibration value determined by the ADC self-calibration
ADCx_CLPX	CLPX Calibration Value The capacitor C10 calibration value determined by the ADC self-calibration
ADCx_CLP9	CLP9 Calibration Value The capacitor C9 calibration value determined by the ADC self-calibration
ADCx_CLPS_OFS ADCx_CLP3_OFS ADCx_CLP2_OFS ADCx_CLP1_OFS ADCx_CLP0_OFS ADCx_CLPX_OFS ADCx_CLP9_OFS	The pre-defined capacitor individual offset correction value for capacitor CS/C14/C13/C12/C11/C10/C9, which is used during the calibration algorithm execution. The user does not need to change these bits as they are for IP internal use.

NOTE

The registers of CLPS/3/2/1/0/X/9, GAIN, OFFSET are the results of one calibration run and are always to be considered as a set. The values belong together.

4. ADC Calibration Software Guidelines

4.1. ADC calibration steps

To run the ADC calibration and ensure that the calibration results are correct, the following steps are required:

- On each reset, wait until reference voltage (VREFH) has stabilized
- Calibrate only one ADC instance at a time. When calibrating instance ADC0, the instances ADC1, ADC2, and so on should be left idle
- Set ADCK (ADC clock) to half the maximum specified frequency. For example: 25 MHz
- Start ADC calibration by writing ADC_SC3 register with: CAL=1, AVGE=1, AVGE=11
- Wait for calibration to finish. This will be indicated by conversion complete flag (COCO in ADC_SC1n)
- Optionally do a calibration check as described below
- Run the ADC conversions with high accuracy in your application. Ensure to re-configure the ADCK clock speed and to re-configure AVGE and AVGS to your desired settings. It is possible to run at maximum clock speed while not using hardware averaging.

4.2. ADC calibration flow chart

The calibration flow chart is shown in Figure 6. The calibration results can be affected by the clock source, frequency, power and conversion speed settings, voltage reference, and hardware average function.

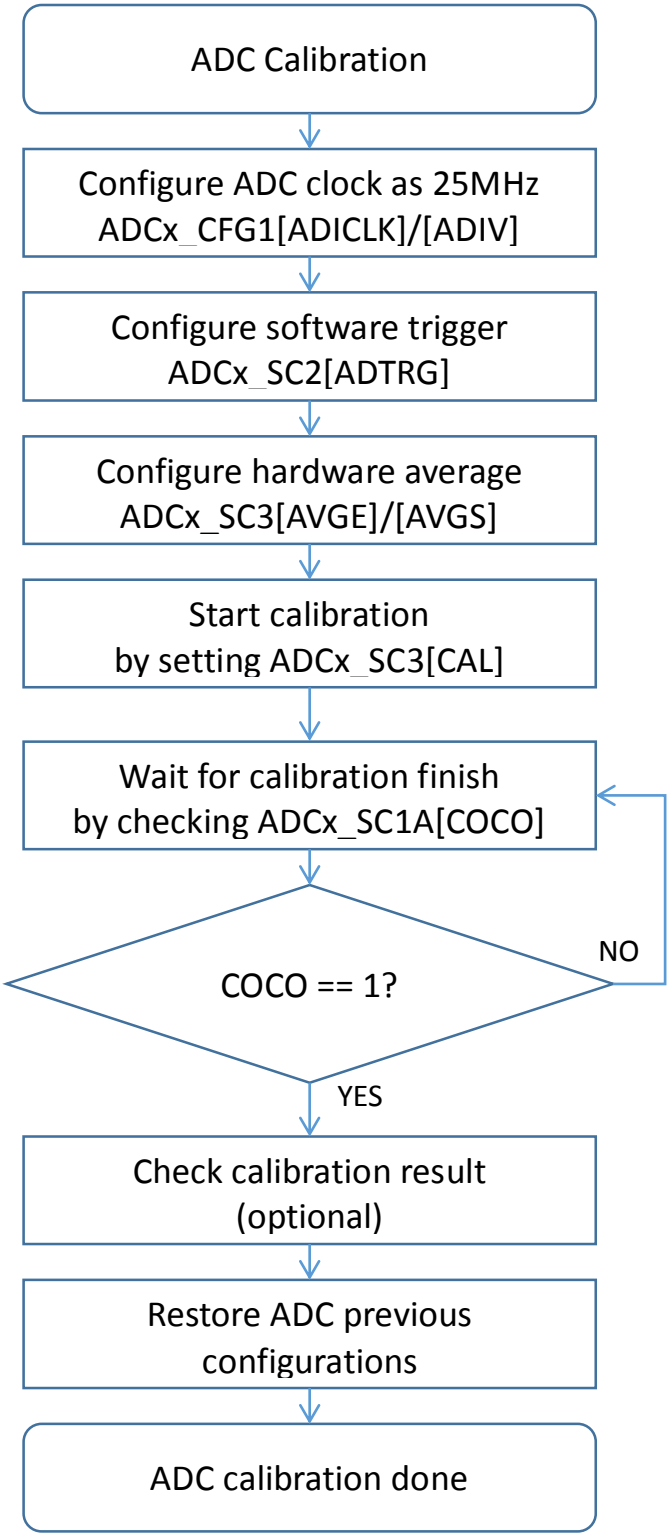

Figure 6. ADC calibration flow chart

4.3. ADC calibration result check

After calibration has finished (COCO=1) the application software can check how well the ADC is set up and integrated on your application board (PCB). For this the register values of ADC_CLP9, ADC_CLPX, ADC_CLPS, ADC_CLP0, ADC_CLP1, ADC_CLP2, and ADC_CLP3 must be read and compared as shown in Table 2.

Table 2. **ADC calibration result check**

Value	min	typical	max
ADC_OFS	-48	-8	22
ADC_CLP9	-12	4	20
ADC_CLPX	-16	0	16
ADC_CLPS	30	72	120
ADC_CLP0 - ADC_CLPS	-14	0	14
ADC_CLP1 - (2*ADC_CLP0)	-16	0	16
CLP2 - (2*ADC_CLP1) + 26	-20	0	20
CLP3 – (2*CLP2)	-36	0	36

4.4. ADC calibration software example

A software example is located in the SDK 2.0 folder boards\twrke18f\driver_examples\adc12, which can be run on TWR-KE18F board, the IDE is IAR 7.6. The calibration register values are shown in Figure 7:

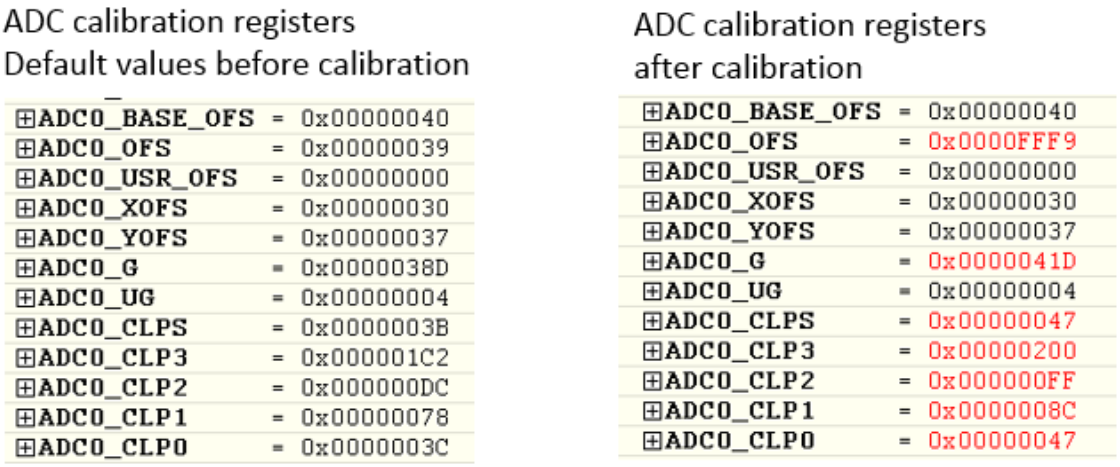

Figure 7. ADC calibration register values

5. ADC Calibration Hardware Guidelines

For high accuracy of ADC (as specified in data sheets KE15Z and KE18F) on the application board (PCB) the following requirements should be met:

- Bypass caps between VREFH and VREFL. Suggested cap sizes: 1nF, 100nF, 1uF
- Place caps on PCB as close as possible to the device pins VREFH and VREFL
- Bypass caps between VDDA and VSSA. Suggested cap sizes: 1nF, 100nF, 1uF
- Place caps on PCB as close as possible to the device pins VDDA and VSSA
- Routing of VDDA, VSSA, VREFH, VREFL on PCB
  - Low impedance between the bypass caps and the MCU pins
  - Keep routing distant from noisy signal routes like switching I/Os.

6. References

The following references are available on the NXP website.

1. *ADC16 Calibration Procedure of MC9S08GW64* (document AN4168)
2. *How to Increase ADC Accuracy in an Application* (document AN5250)
3. *16-bit SAR ADC Calibration*. See the NXP Community.

7. Revision History

Table 3. **Revision history**

Revision	Date	Substantive changes
1	08/2016	Table 1: removed “value range” column
0	07/2016	Initial release

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number: AN5314
Rev. 1
08/2016