Analog-to-Digital Converter in the SAM3S4

1. Introduction

The purpose of this application note is to describe, from a software point of view while considering analog constraints, correct usage of the ADC implementation in the SAM3S4 series of the Atmel family of Arm®, Cortex™-M3 based Flash MCUs. The first objective is to define the precise timing setup of the ADC to avoid inexact usage that can lead to erroneous voltage conversion.

Analog-to-Digital converters (ADC) translate analog measurements, characteristic of most phenomena in the real world, to digital format to be used in information processing, computing, data transmission, and control systems.

In order to better understand the cyclic pipelined ADC settings, a brief description of the ADC is presented in order to more fully comprehend a voltage conversion. (See Section 3. on page 3.)

Offset and gain error parameters of this ADC can be improved by calibration. Calibrating offset and gain error is defined in Section 5. “ADC Calibration”.

AT91SAM ARM-based Flash MCU

Application Note

11106A–ATARM–01-Jul-11

2. Terminology

Terminology pertinent to ADC is provided below in Table 2-1 and organized by category.

Table 2-1. ADC Terminology

Code	Description
Related to Numeric Base	
0b	Binary Notation
0x	Hexadecimal Notation
Related to Application	
SOC	Start of Conversion
m	Number of steps to perform a conversion
N	ADC output numbers of bits
Tconv	Conversion time of the ADC, expressed in number of ADC clock cycle
Fadc	ADC clock cycle, obtained from the product master clock
Tcp_adc	Period of an ADC clock frequency, Tcp_adc = 1/Fadc
Fs	Sampling frequency on the ADC input voltage for one channel.
Track	The tracking time is the necessary time the ADC must see the input voltage on its input in order to reach the specified accuracy.
Related to Electrical	
LSB	Least Significant Bit
Zsource	Zsource is an external voltage source impedance, supplying the voltage to the ADC

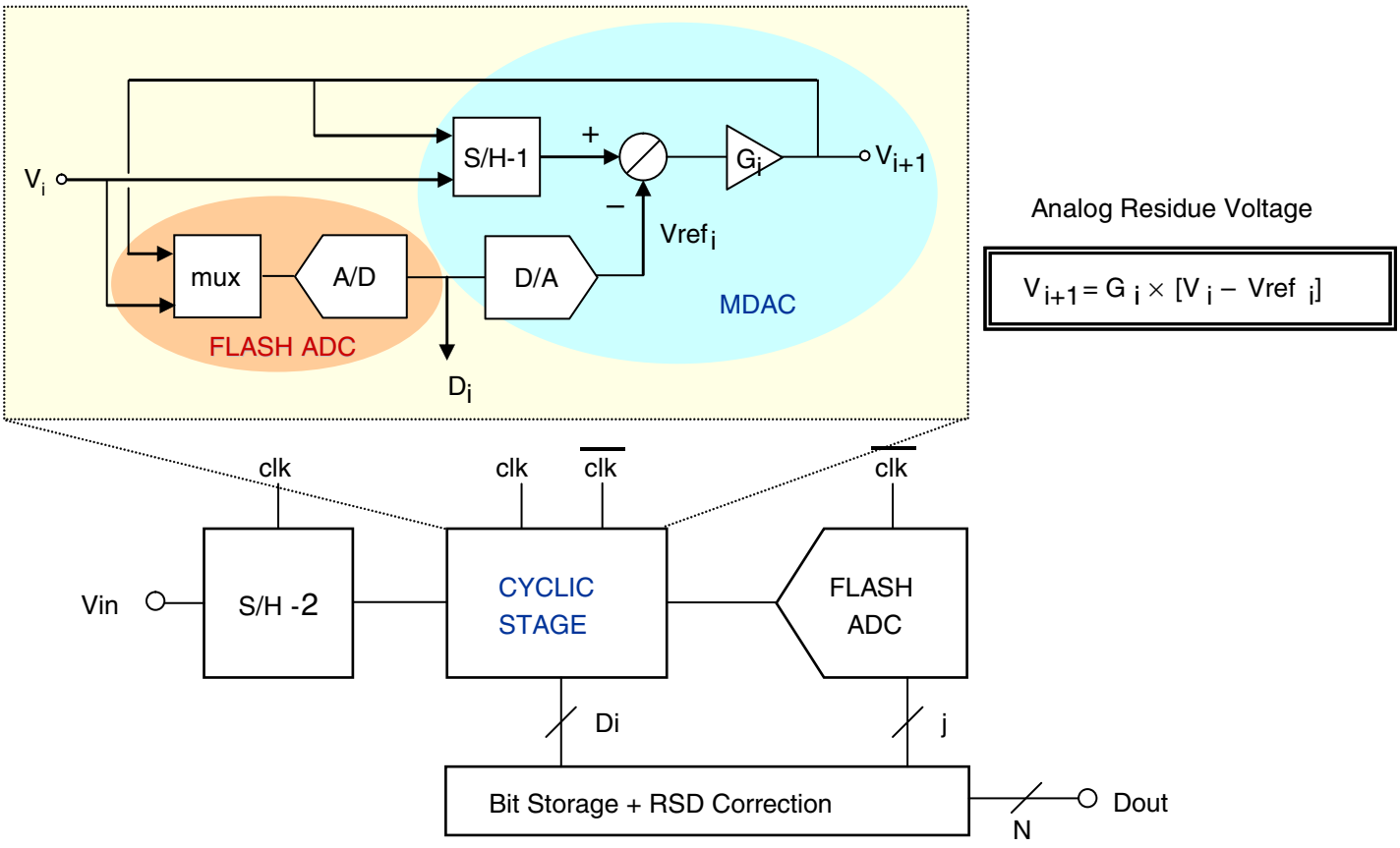
3. General Information

Atmel developed a specific ADC for the SAM3S4 product family, where the target was to reach 12 bits accuracy and a data rate of 1 Ms/S. This was made possible with the cyclic pipeline ADC technique (also called algorithmic ADC). The cyclic pipeline ADC is a technique different from that of the SAR (Successive Approximation Register) ADC.

3.1 Cyclic ADC Converter

The principle of the conversion is to find a set of reference voltages whose sum equals the signal sample being converted. This is done by sequentially subtracting different reference voltages from the input sample until the residue voltage becomes less than the desired LSB, indicating that the sum of the subtracted references equals the original sample value.

Figure 3-1. 12-bit ADC Detailed Block Diagram



In a pipelined ADC, the residue is amplified between subtraction steps in order to increase accuracy.

For each pipeline step “i = 1 to m”: $Vi + 1 = Gi \times (Vi - D i \times Vref)$

Where: Gi is a gain of 2, Di is a digital code from the 1.5 bits ADC, Vref is a reference voltage.

In a cyclic pipelined ADC architecture (see Figure 3-1), only one stage is used several times to serially produce the digital bits, Di. All Di data are combined into the digital part of the ADC to form the N bits of the output word, Dout. This process includes bit redundancy to perform some

error correction. The analog residue voltage is recycled m times as long as the resolution is not achieved.

To build a 12-bit converter (N = 12), a full conversion operates in a sequence of m = N-1 = 11 steps. A total of 15 ADC clock cycles are necessary to perform this conversion. Five extra clock cycles are necessary to sample and hold (S/H-2) the input voltage before conversion. During that time the ADC input is high impedance. In total Tconv = 20 ADC clock cycles are used to perform a conversion.

3.2 Sample and Hold Circuit

The S/H-2 block buffers the input voltage to be converted and transfers it to the ADC core.

This block brings an advantageous feature to this ADC, because a new sample can be tracked during the conversion time (15 ADC clock cycles), no time is wasted for a new tracking period.

4. ADC Timing

4.1 ADC Mode Register Settings

Some SAM3S4 ADC Mode Register (ADC_MR) settings are described in this section.

4.1.1 TRANSFER

The TRANSFER value can be coded on 2 bits. The transfer period is the time necessary to transfer the sampled input analog data to the conversion circuitry of the ADC, during that time the input of the ADC is high impedance (no input voltage tracking possible).

Transfer time = (TRANSFER × 2 + 3) of ADC clock period (Tcp_adc).

- Transfer time Max = 11 Tcp_adc
- Transfer time Min = 3 Tcp_adc

A channel modification (ex: ch0 to ch1) always occurs after the Transfer time. In the case of the SAM3S4 cyclic pipeline ADC, the Transfer time = 5 Tcp_adc after SOC fixed by the ADC design.

If a higher value is programmed, then the channel modification is delayed and reduces the tracking time.

If a smaller value is programmed then the channel modification occurs during the transfer time.

By consequence, the optimal value for SAM3S4 is TRANSFER = 1, programmed Transfer time = 5 × Tcp_adc.

4.1.2 TRACKTIM

The value of TRACKTIM is coded on 4 bits, Tracking Time = TRACKTIM × Tcp_adc periods.

This time is used to track the input channel with the necessary accuracy. The value can go from 0 to 15 Tck. In the SAM3S4 product, this time is placed between ADC conversions. As this cyclic pipeline ADC can sample tracked voltage during a conversion. It is not necessary to use the TRACKTIM time defined in the ADC_MR register. For more information see the ADC: Functional Description in the SAM3S4 datasheet.

By consequence, the optimal value for SAM3S4 is TRACKTIM = 0. The real tracking time is discussed later in this application note. See: Section 4.3 “Tracking Time”.

4.1.3 STARTUP

The value is coded on 4 bits, Startup time = 0×Tcp_adc to 960×Tcp_adc with 16 steps not linear (for intermediate values see the STARTUP bit field description in the ADC Mode Registers section of the SAM3S4 datasheet).

The startup time depends on the ADC clock speed =1/Tcp_adc, see Table 4-1:

Programming the startup time also depends on the SLEEP mode or FAST Wake up mode (FWUP).

- The ADC needs 40 µS to go from sleep to normal mode
- The ADC needs 12 µS to go from fast wake up to normal mode

Below, Table 4-1 columns SLEEP and FWUP compute the number of ADC clock cycles to match the ADC startup time requirement respectively of 40 µS and 12 µS.

Master clock MCK and PRESCALE are provided as examples, for other values STARTUP must be recomputed following the same methodology.

The columns STARTUP for SLEEP and STARTUP for FWUP give the closest expected value to match the ADC startup time. The result is not optimum because there are no intermediate values between 112 and 512, and none between 24 and 64.

Table 4-1. STARTUP Value

MHz		MHz	Nb of Tcp_adc	Nb of Tcp_adc	Nb of Tcp_adc	Nb of Tcp_adc
MCK	PRESCAL	ADC clock Fadc	SLEEP	FWUP	STARTUP for SLEEP	STARTUP for FWUP
40	0	20	800	240	832	512
64	1	16	640	192	640	512
64	2	10.67	427	128	512	512
64	3	8	320	96	512	96
64	4	6.40	256	77	512	80
64	5	5.33	213	64	512	64
64	6	4.57	183	55	512	64
64	7	4	160	48	512	64
64	8	3.56	142	43	512	64
64	9	3.20	128	38	512	64
64	10	2.91	116	35	512	64
64	11	2.67	107	32	112	64
64	12	2.46	98	30	112	64
64	13	2.29	91	27	96	64
64	14	2.13	85	26	96	64
64	15	2	80	24	80	24
32	15	1	40	12	64	24

When ADC goes into SLEEP mode between successive conversions (to save on power consumption), the sampling frequency (Fs) will be strongly affected by the ADC startup.

- SLEEP mode: Fs = Fadc/(STARTUP for SLEEP +Tconv), Tconv = 20 clock cycles
- FWUP mode: Fs = Fadc/(STARTUP for FWUP + Tconv), Tconv = 20 clock cycles

Table 4-2. ADC Sampling Frequencies

MHz		MHz	Nb of Tcp_adc	Hz	Nb of Tcp_adc	Hz
MCK	PRESCAL	ADC clock: Fadc	STARTUP for SLEEP mode	FS due to SLEEP mode	STARTUP for FWUP mode	FS due to FWUP mode
40	0	20	832	23474	512	37594
64	1	16	640	24242	512	30075
64	2	10.67	512	20050	512	20050

Table 4-2. ADC Sampling Frequencies (Continued)

MHz		MHz	Nb of Tcp_adc	Hz	Nb of Tcp_adc	Hz
MCK	PRESCAL	ADC clock: Fadc	STARTUP for SLEEP mode	FS due to SLEEP mode	STARTUP for FWUP mode	FS due to FWUP mode
64	3	8	512	15038	96	68966
64	4	6.40	512	12030	80	64000
64	5	5.33	512	10025	64	63492
64	6	4.57	512	8593	64	54422
64	7	4	512	7519	64	47619
64	8	3.56	512	6683	64	42328
64	9	3.20	512	6015	64	38095
64	10	2.91	512	5468	64	34632
64	11	2.67	112	20202	64	31746
64	12	2.46	112	18648	64	29304
64	13	2.29	96	19704	64	27211
64	14	2.13	96	18391	64	25397
64	15	2	80	20000	24	45455
32	15	1	64	11905	24	22727

4.1.4 SETTLING

The value is coded on 2 bits. Settling time = (SETTLE+1) × Tcp_adc

Settling time is the time needed to stabilize the ADC when modifying the gain and offset (static working condition).

- When Fadc = 20 MHz then SETTLING = 3.
- When Fadc = 10 MHz then SETTLING = 1.
- When Fadc ≤ 5 MHz then SETTLING = 0.

It is recommended to set a default value of SETTLING = 3 to ensure proper stabilization of the ADC.

4.1.5 PRESCAL

The PRESCAL is used to set the ADC frequency from the master clock MCK to a value targeted by the customer application: Fadc = MCK / ((PRESCAL+1) × 2) = 1/Tcp_adc. The Sampling frequency on the input voltage depends on the ADC clock, the number of channels, and the ADC conversion time (tconv = 20 × Tcp_adc). Cycling through channel will lose one ADC clock cycle to check for offset and gain modification for each loop.

4.2 ADC Mode of Operation: Freerun or Triggered ADC

The sampling rate is optimized in Freerun mode and does not allow SLEEP or FWUP mode.

The sampling rate is reduced in Triggered ADC mode to allow down sampling and/or accept longer tracking time.

Freerun mode is suitable to reach the highest sampling rate. In this case the tracking time of the input voltage is defined by the conversion time of the ADC, 15 × Tcp_adc. By default the ADC is in tracking mode.

A triggered ADC is used to obtain any one of the following:

- to increase the tracking time
- to put the ADC in SLEEP or FWUP mode between samples in order to save energy
- to reduce the sampling rate

Regardless of the triggering method (software or hardware), the sampling rate is defined by the trigger frequency.

4.3 Tracking Time

4.3.1 Track Time ≤15 × Tcp_adc: Freerun Mode

After specifying MCK/PRESCAL/ADC_clock for a targeted sampling datarate, the source impedance of the external voltage to sample will have a maximum allowable value to guaranty a proper conversion.

In 12-bit mode: Ttrack = 0.054 × Zsource + 205 = Tracking time (nS)

Table 4-3. External Voltage Source Maximum Impedance

MHz		MHz	kHz or Ks/S	nS	K Ohms
MCK	PRESCAL	ADC clock	ADC DATARATE	Tracking time	Zsource 12 bits
40	0	20	1000	750	10
64	1	16	800	938	14
64	2	10.67	533.33	1406	22
64	3	8	400	1875	31
64	4	6.40	320	2344	40
64	5	5.33	266.67	2813	48
64	6	4.57	228.57	3281	57
64	7	4	200	3750	66
64	8	3.56	177.78	4219	74
64	9	3.20	160	4688	83
64	10	2.91	145.45	5156	92
64	11	2.67	133.33	5625	100
64	12	2.46	123.08	6094	109
64	13	2.29	114.29	6563	118
64	14	2.13	106.67	7031	126
64	15	2	100	7500	135
32	15	1	50	15000	274

4.3.2 Track Time ≥ 15 × Tcp_adc: TRIGGERED Mode

The computation of the new tracking time depends mainly on the value of Zsource (if Zsource ≥ 274 kOhm) and is higher than the conversion time of the ADC. The needed additional time is then given by the SAM3S4 timer and software/hardware triggering function.

The maximum, possible sampling frequency is defined by the Ttrack time in nano seconds, computed by formula above, minus the 15 Tcp_adc and plus TRANSFER time.

In 12-bit mode:

- 1/Fs = Ttrack - 15 × Tcp_adc + 5 ×Tcp_adc

Reducing power consumption is another motivation to trigger the ADC. Therefore, it is advantageous to turn off the ADC between successive conversions.

If the ADC is turned off during sampling, then the sampling rate may be modified again. Refer to the SLEEP/FWUP mode constraint (see: Section 4.1.3 “STARTUP”).

5. ADC Calibration

5.1 Offset and Gain Calibration Introduction

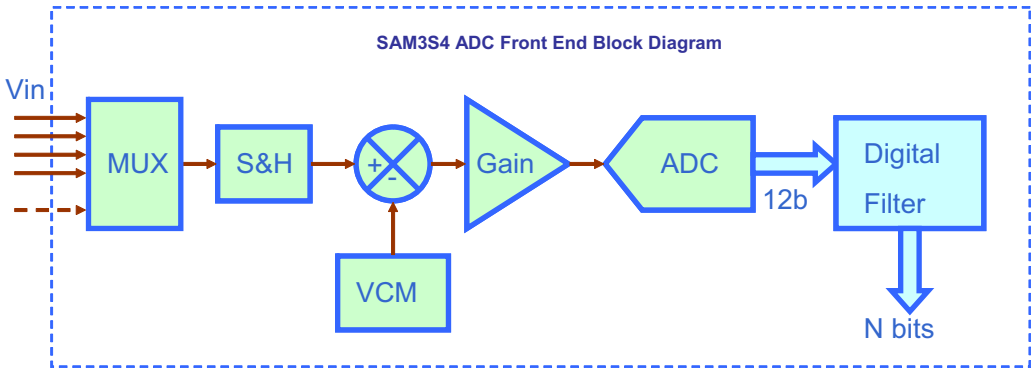
A signal measurement can be significantly affected by offset or gain error. These errors come from device process deviation and are subject to process variation and temperature. The impact of these errors depends on the amplitude of the input signal versus the offset or gain error for a given expected output resolution.

A typical measurement system may consist of the following:

- Sensor source, Vin
- Reference voltage Vrefin, The common mode voltage (VCM) is usually Vrefin/2
- Gain to amplify the sensor signal
- SAM3S4 12-bit ADC

The gain error and offset error come from the VCM, sample & hold and gain block inaccuracies

Figure 5-1. SAM3S4 ADC Front End Block Diagram



5.2 Definitions

5.2.1 Absolute Accuracy Computation

For a given list of errors: err1, err2, …, errn, with associated statistical data:

- Standard deviation: std_err1, std_err2, …, std_errn
- Averaged values: avg_err1, avg_err2, …, avg_errn

Computation of the final error average:

- Avg_err = avg_err1 + avg_err2 + …+ avg_errn

Note: Most of the time the average error is at zero.

Computation of the final error standard deviation (1sigma):

- Std_err = Square_Root_of{(std_err1)^2 + (std_err2)^2 +…+ (std_errn)^2}

3 sigma total error computation from averaged value:

- Absolute accuracy = Avg_err+/- 3× Std_err

Absolute accuracy is also called Absolute error.

5.2.2 Accuracy or Error Computation Relative to Input (RTI)%

The accuracy of the ADC is defined as the square root of the sum of all squared errors divided by the signal amplitude.

Accuracy RTI(%) =1 00 × Total_error/(Vin×Gain)

- Accuracy RTI(%) = Zero is the targeted value
- Accuracy RTI(%) = 0.1% is about 9.5 bits resolution (-60 dB)
- Accuracy RTI(%) = 0.01% is about 13 bits resolution (-80 dB)

5.3 Motivation for Calibration

Absolute accuracy and the accuracy relative to the input are two essential parameters to consider in system performance. The computation is performed with the offset error, the gain error, Integral Non Linearity (INL) error, Voltage reference spread and temperature dependency.

This is illustrated through an example of offset error on the measurement accuracy.

5.3.1 Gain = 1 in Front of the ADC

In this case, only the offset error is considered. The gain error alone gives the same conclusion and other errors such as INL are neglected.

The ADC setup is: Vrefin = 3V, Single mode with ADC gain=1, input voltage range is from 0 to Vrefin

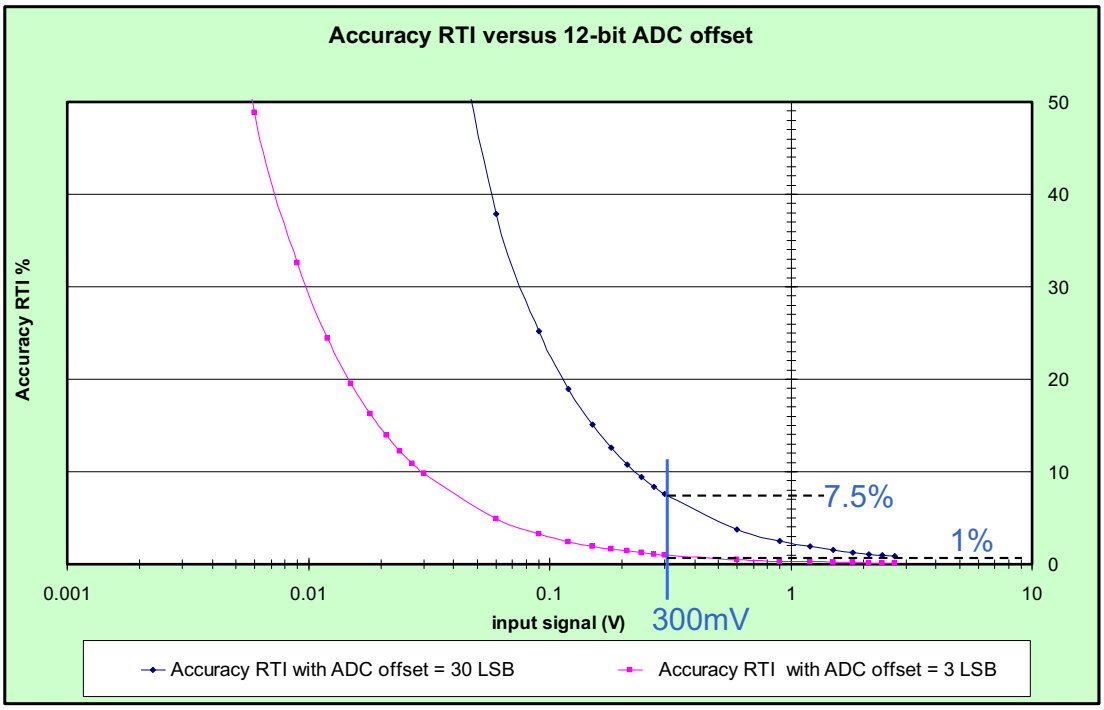
- Offset_error = 30 LSB for a non calibrated ADC, with LSB=3/4096 Volt
- Offset_error = 3 LSB for a calibrated ADC, with LSB=3/4096 Volt

Then,

- Accuracy RTI(%) = 100 × Offset_error/VIN

Below, Figure 5-2 illustrates the loss of accuracy when the offset has two different values.

Figure 5-2. Loss of Accuracy



It is obvious that calibrating the ADC will have a great improvement on the measurement, especially when the input signal is small.

For 300 mV input signal and if a calibration is performed on this ADC, the accuracy will be improved from 7.5% down to 1%.

5.3.2 Amplification in Front of the ADC:

If a gain>1 is applied:

- Accuracy RTI(%) = 100 × Offset_error/(VIN×Gain)

The offset error is divided by the gain, so the accuracy is improved and the ADC offset becomes negligible. In this case, calibration is not necessary.

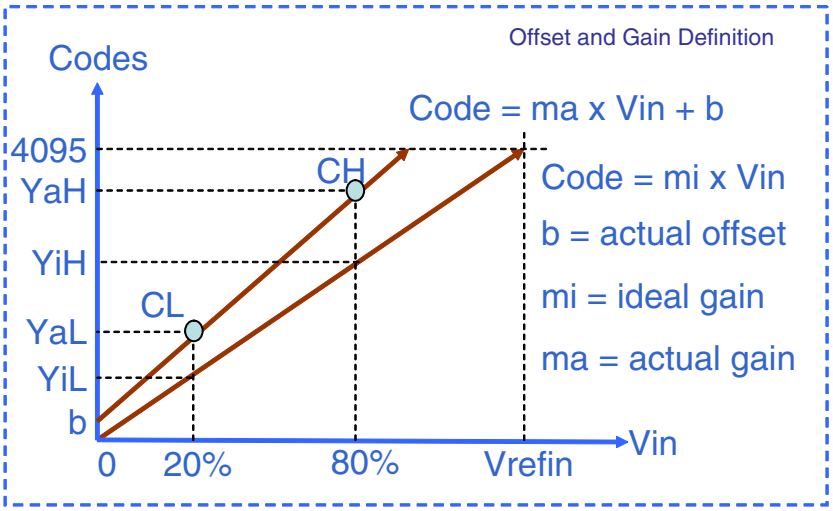
For 300 mV input signal. The accuracy will be improved from 7.5% down to 1% if a gain of 10 is applied on this ADC. The internal gain of the ADC is limited to 4, an additional external gain of 2.5 is used.

5.4 Calibration of the ADC

It is necessary to calibrate both gain and offset of the ADC. Starting with the offset calibration and finishing with the gain calibration. This calibration must be performed for a specific gain, Vrefin, (single or differential) and offset setup of the ADC. If several conditions are used in an application then a calibration is needed for each.

5.4.1 Offset and Gain Error Definition

Figure 5-3. Offset and Gain Error Definition



Calibration is performed by feeding two known reference values into the ADC. This method is the same used in the factory production test to extract the data used for software calibration. This calibration is relative to the voltage reference supplied to the ADC.

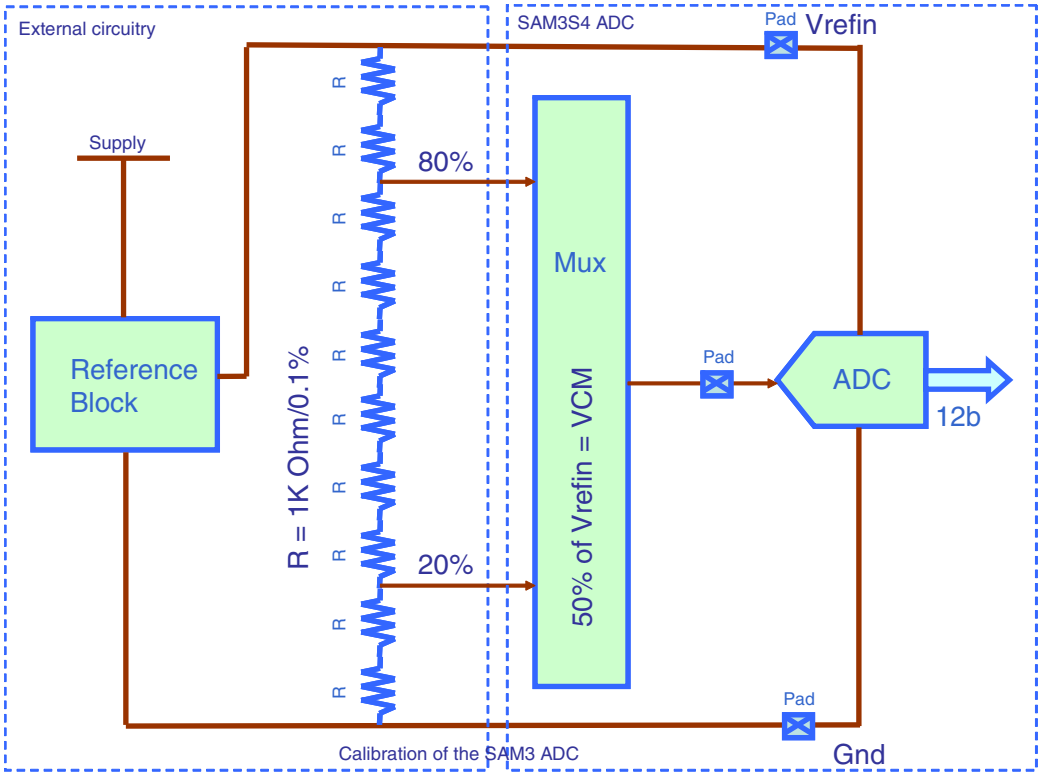
- CH is highest actual point measured with Vin = 80% of Vrefin.
- CL is lowest actual point with Vin = 20% of Vrefin.

Points	Actual	Ideal
CH	YaH	YiH = 3276
CL	Yal	Yil = 819

From these two points, the software can compute the calibration of gain and offset to compensate the ADC errors.

How the input voltage references are generated is shown in Table 5-4 below. The Vrefin can also be connected to the 3.3V supply. To avoid noise issues in the measurement of the two points CL and CH, an average of 256 samples is performed, this reduces the noise by a factor 16.

Figure 5-4. Input Voltage Reference Generation



5.4.2 Calibration Process

The ideal calibration equation is as follows:

- Corrected_code = mc × (Actual_code + bc) = mc × Actual_code + mc × bc

Where, mc is the factor for a gain correction, and bc is the offset correction.

The calibration computation should be as easy as possible and use the minimum clock cycle of the Cortex M-3. The preferred instruction for this is a MAC instruction. First perform a multiplication and second an addition, in only one clock cycle.

So the equation for the calibration must be:

- Corrected_code = mc × Actual_code + bc

In this case a negligible error is performed on the calibration, mc × bc is simplified to bc.

Computation of the gain correction mc:

The tester will perform this computation from the two measured points CL and CH.

Then:

- mc = (YiH-YiL)/(YaH-YaL)

The Gain correction factor mc is converted into Fixed Point format and stored into the SAM3S4 memory. (See: ADC Electrical Characteristics in the SAM3S4 datasheet.)

Fixed Point Format

Integer 16 bits																Fractional part 16 bits															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

For example:

- The ratio mc = 0.989130435

Then Compute the integer of the mc × 2^16 and convert it to Hexadecimal. The product memory will store the value of mc = 0x0000FD37 for calibration.

Computation of the offset correction bc:

- bc = YiH-YaH*mc

The offset correction factor is exactly the offset value coded in Fixed Point format and Two's complement for negative values.

- bc = -b × 2^16

For example:

The offset b=45 is 2949120 in Fixed Point decimal format for bc = 0xFFD30000 in hexidecimal, Fixed Point format.

If the offset coefficient factor is negative, then it will be coded as Fixed Point format, Two's complement.

As the ADC conversion is 12 bits, the tester stores the data in memory, keeping the last MSB at 0. A function must be used to convert those numbers into the correct 16-bit format:

code example:

```
/*Convert 12 bits to 16 bits signed */
void C12bits_to16bits_conversion( unsigned int *ui_offset)
{
  /* Test if 12bits word is positive or negative and adjust it.... on 32 word
bits*/
  if ((*ui_offset& 0x08000000)==0x08000000)
  {
  *ui_offset|=0xF0000000; /*negative value*/
  }
  else
  {
  *ui_offset&=~(0xF0000000) ; /*positive value*/
}
return;
}
```

The memory will store the value 0x0FD30000.

The above function will produce the value to use in the calibration: 0xFFD30000.

5.4.3 How to Compute the Calibration

Apply the Fixed Point formula as follows:

- Corrected_code = mc × Actual_code + bc

The output result is Fixed Point format with fractional part from bit0 to bit15 and integer part from bit16 to bit27.

Example in hexadecimal for an ideal code of 0x07E80000, (2024 in decimal):

Gain correction factor mc = 0.989130435

Integer 16 bits				Fractional part 16 bits			
0	0	0	0	F	D	3	7

Offset correction bc = -45

Integer 16 bits				Fractional part 16 bits			
F	F	D	3	0	0	0	0

The actual ADC code is 2091 (decimal) 0x0000082B

ADC Code				ADC Code			
0	0	0	0	0	8	2	B

The result of mc × Actual_code + bc

Integer 16 bits				Fractional part 16 bits			
0	7	E	7	0	0	0	0

The Corrected code is then 0x07E70000. The ideal code is 0x07E80000, there is an error of 1 LSB in the correction.

Revision History

Doc. Rev	Comments	Change Request Ref.
11106A	First issue	

Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: (+1) (408) 441-0311
Fax: (+1) (408) 487-2600

International

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
JAPAN
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com
www.atmel.com/AT91SAM

Technical Support
AT91SAM Support
Atmel technical support

Sales Contacts
www.atmel.com/contacts/

Literature Requests
www.atmel.com/literature

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© 2011 Atmel Corporation. All rights reserved. Atmel®, Atmel logo and combinations thereof, DataFlash@ and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. ARM®, ARMPowered® logo, and others are registereed trademarks or trademarks of ARM Ltd. Other terms and product names may be trademarks of others.