TEXAS INSTRUMENTS

Linear Improvement of the MSP430 14-Bit ADC Characteristic

Application Report

June 1999

Mixed Signal Products

SLAA048

Contents

List of Figures

List of Tables

Linear Improvement of the MSP430 14-Bit ADC Characteristic

Lutz Bierl

ABSTRACT

This application report shows different linear methods to improve the accuracy of the 14-bit analog-to-digital converter (ADC) of the MSP430 family. Different correction methods are explained: some with monotonicity and some using linear regression. The methods used differ in RAM and ROM allocation, calculation speed, reachable improvement, and complexity. For all correction methods, proven, optimized, software examples are given with 8-bit and 16-bit arithmetic. The *References* section at the end of the report lists related application reports in the MSP430 14-bit ADC series.

1 Introduction

The application report *Architecture and Function of the MSP430 14-Bit ADC*[1] gives a detailed overview to the architecture and function of the 14-bit analog-to-digital converter (ADC) of the MSP430 family. The principle of the ADC is explained and software examples are given. Also included are the explanation of the function of all hardware registers contained in the ADC.

The application report *Application Basics for the MSP430 14-Bit ADC*[2] shows several applications of the 14-bit ADC of the MSP430 family. Proven software examples and basic circuitry are shown and explained.

The application report *Additive Improvement of the MSP430 14-Bit ADC Characteristic*[3] explains the external hardware that is needed for the measurement of the characteristic of the MSP430's analog-to-digital converter. This report also demonstrates correction methods that use only addition. This allows the application of these methods in real time systems, were execution time can be critical.

Figure 1 shows the block diagram of the 14-bit analog-to-digital converter of the MSP430 family.

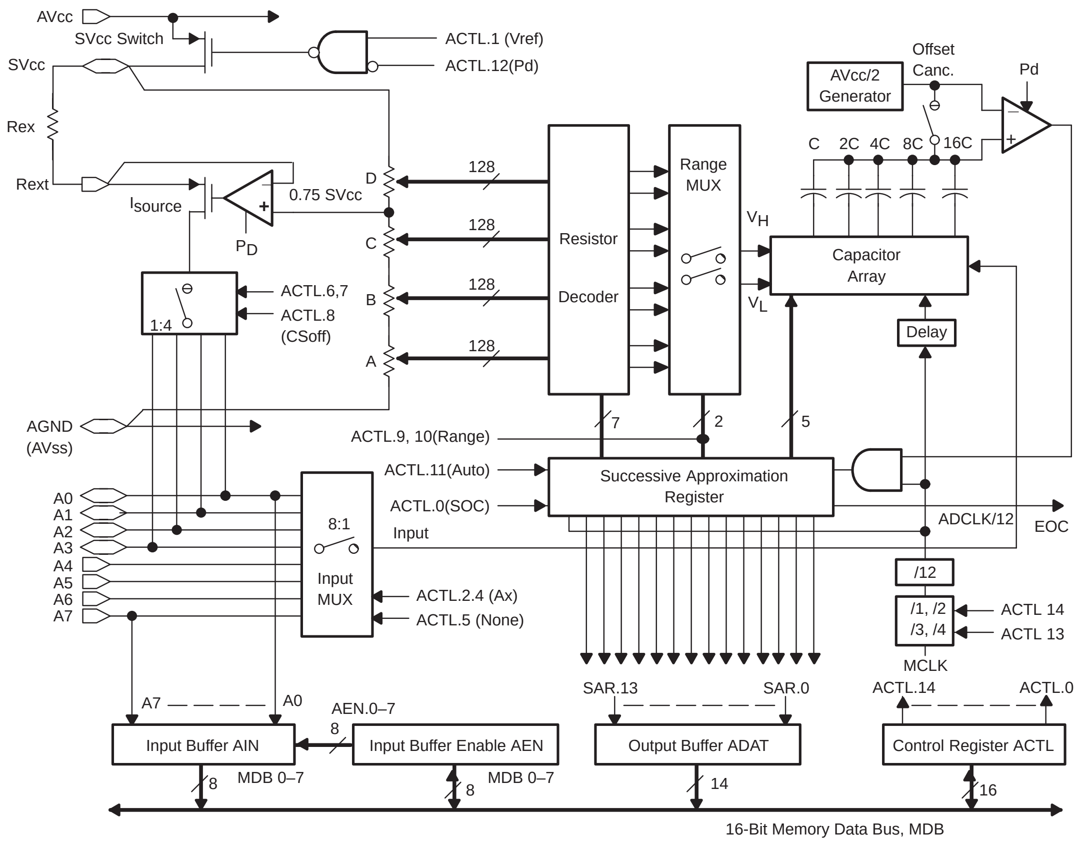


Figure 1. The Hardware of the 14-Bit Analog-to-Digital Converter

The methods for the improvement of the ADC described in the next sections are:

- Linear equations with border fit: single linear equation per range
- Linear equations with border fit: multiple linear equations per range
- Linear equations with linear regression: single linear equation per range
- Linear equations with linear regression: multiple linear equations per range

Quadratic and cubic corrections are explained in the application report *Nonlinear Improvement of the MSP430 14-Bit ADC Characteristic*[4].

1.1 Correction With Linear Equations

A good error correction with low RAM requirements is possible if not only the offset error—like with the additive methods—but also the slope error of the ADC characteristic can be corrected. However, this requires the use of a multiplication. The multiplication subroutine used here is is optimized for real time environments: it terminates immediately after the unsigned operand—the ADC result—becomes zero due to the right shift during the multiplication. The subroutine is appended to the first software example (see section 1.2.1.1). The full code with explanations and timing is contained in *Nonlinear Improvement of the MSP430 14-Bit ADC Characteristic*, SLAA050[4].

The generic correction formula for linear correction, which is valid for floating point or 16-bit integer arithmetic, is:

$$Ni_{corr} = Ni + (Ni \times a1 + a0)$$

The optimized 16-bit multiplication subroutine for the above formula—including the calculation software—is included in section 1.2.2.1, *Linear Regression: Single Linear Equation per Range*. The full code is described in the *MSP430 Application Report*[5], *Integer Calculation Subroutines* section of Chapter 5.

The floating point example given for the cubic correction—see *Nonlinear Improvement of the MSP430 14-Bit ADC Characteristic*[4]—may be adapted easily to the calculation of linear equations: the unused terms—the quadratic and cubic terms—are simply left out.

The formulas to calculate the correction coefficients a1 (slope) and a0 (offset) out of the two known errors e2 and e1 of the ADC steps N2 and N1 are:

$$a1 = -\frac{e2 - e1}{N2 - N1} \qquad a0 = -\frac{e1 \times N2 - e2 \times N1}{N2 - N1}$$

The advantages of the negated correction coefficients a1 and a0 are:

- Shorter and faster software: the INV (invert) and INC (increment) instructions for the negation of the corrections are not necessary
- The ADAT register (ADC result register) is a read-only register and can be used for additions directly. If the correction needs to be subtracted from the ADAT register, then an intermediate step is necessary.

All principle figures of this report—as in *Additive Improvement of the MSP430 14-Bit ADC Characteristic*[3]—have the same structure:

- The black straight line indicates the negated correction value (this is to show the precision of the correction).
- The scribbled black line indicates the noncorrected ADC characteristic.
- The white line shows the corrected ADC characteristic.
- The small circles indicate the measured ADC points (not all measured samples are shown).

An example using the 16-bit arithmetic is given in section 1.2.2.1, *Linear Regression: Single Equation per Range*.

All other given equations in the following sections assume the use of the 8-bit arithmetic as described in *Nonlinear Improvement of the MSP430 14-Bit ADC Characterisitc*,SLAA050[4]. Therefore the correction formulas are adapted to the limited, but fast 8-bit arithmetic. This reduced arithmetic makes relative addresses for the ADC steps necessary: the full ADC range is divided into sections and the ADC value is adapted to 128 subdivisions for the full section. The equations for the 8-bit arithmetic are given and explained with each method.

1.2 Coefficients Estimation

With the maximum possible ADC error (±10 steps contained in a band of ±20 steps) the maximum values for the coefficients a1 and a0 are:

Table 1. Worst Case Coefficients With 8-Bit Arithmetic

EQUATIONS PER RANGE	SINGLE	TWO	FOUR
Linear coefficient a1:	±0.15625	±0.078125	±0.0390625
Format a1 (integer.fraction)	0.9	0.10	0.11
Constant coefficient a0:	±20.00	±20.00	±20.00
Format a0 (integer.fraction)	5.2	5.2	5.2
Sections (Equations) per Range	1	2	4
Subdivisions per Range	128	2 × 128	4 × 128
Maximum Change within Section	±20 Steps	±10 Steps	±5 Steps

The above maximum coefficients occur for a single equation per range when the ADC error changes 20 steps within an ADC range (4096 steps) e.g. from +10 to –10 steps or vice versa. For two and four equations per range, the maximum change is appropriately smaller (±10 resp. ±5 steps). This leads to smaller coefficients a1.

- The 8-bit arithmetic operates with signed 8-bit coefficients and an ADC result that is adapted to a value ranging from 0 to 127.
- The 16-bit arithmetic uses the full ADC result (0 to 16383) and signed 16-bit numbers for the calculations.
- The floating point calculation also uses the full ADC result (0 to 16383) and a 32-bit number format for the calculations.

NOTE: Within the software examples, at the right margin of the source code the format of the numbers is noted. The meaning of the different notations is:

0.7	No integer bits, 7 fraction bits. Unsigned number
±4.3	Four integer bits, 3 fraction bits. Signed number
8.0	Eight integer bits, no fraction bits. Unsigned integer number
±7.0	Seven integer bits maximum, no fraction bits. Signed integer number

The statistical results are given separately for the full ADC range (ranges A to D) and for the ranges A and B only. The reason for the second case is the internal current source that is used by many applications: with its use the ADC ranges are restricted to the ranges A, B, and the lower part of range C.

1.2.1 Linear Equations With Border Fit

If monotonicity of the corrected ADC characteristic is a requirement, then the correction methods using the border fit are the right choice. They guarantee, that the four ranges continue smoothly at its borders. This feature is important if the differences of two ADC results are used for calculations.

1.2.1.1 Single Linear Equation per Range

The ADC is measured at the five borders of the four ADC ranges (Ni = 50, 4096, 8192, 12288, 16330). These five results are used for the calculation of the offsets and the slopes of all four ADC ranges.

> **NOTE:** The ADC points 0 and 16383 (3FFFh) including small bands cannot be measured. This is the reason for the use of steps 50 and 16330 in the above explanation.

The formula for the offset a0 and the slope a1 for each one of the four ranges is:

$$Ni_{corr} = Ni + \left[\left(\frac{Ni}{4096} - n \right) \times 128 \times a_1 + a_0 \right]$$

$$a_1 = -\frac{e_u - e_l}{128} \qquad a_0 = -e_l$$

Where:

Ni_{corr}	= Corrected ADC sample	[Steps}
Ni	= Measured ADC sample (noncorrected)	[Steps]
n	= Range number (0...3 for ranges A...D)	
a_1	= Slope of the correction	
a_0	= Offset of the correction	[Steps]
e_u	= Error of the ADC at the upper border of the range	[Steps]
e_l	= Error of the ADC at the lower border of the range	[Steps]

The term $\left(\frac{Ni}{4096} - n \right) \times 128$ of the equation above is the adaptation of a complete section—here a full range—to 128 subdivisions. The calculation of the term is made by simple shifts and logical AND instructions and not a division and a multiplication. See the initialization part of the software example.

The principle of the correction with a linear equation for each range (border fit) is shown in Figure 2. Border fit means, that the borders of the four ranges A to D fit together without gaps from one range to the other one: the border value is used for both ranges.

The improvement methods and their results for this report are demonstrated with the characteristic of device 1 and device 2 due to their worst characteristic compared to the other three devices shown in *Architecture and Function of the MSP430 14-Bit ADC Application Report*.[1]

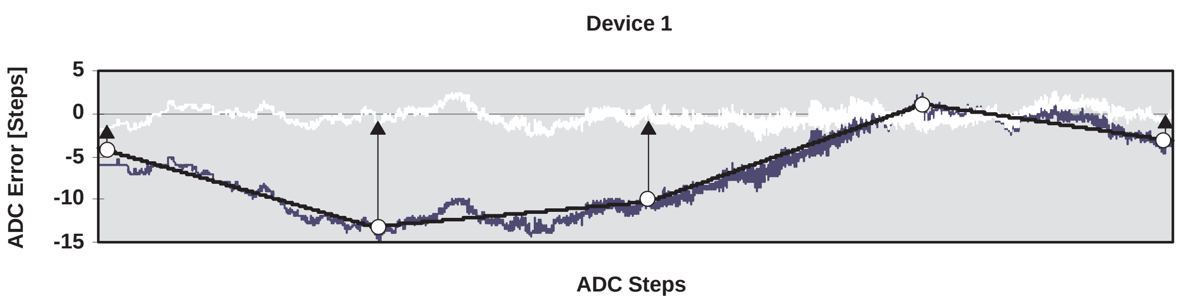


Figure 2. Principle of the Correction With Border Fit (single linear equation per range)

The statistical results for this simple correction method are:

	Full range	Ranges A and B only
Mean Value:	–0.32 Steps	–0.33 Steps
Range:	5.6 Steps	5.1 Steps
Standard Deviation:	0.94 Steps	0.99 Steps
Variance:	0.88 Steps	0.98 Steps

Figure 3 shows the results of this method in a graph.

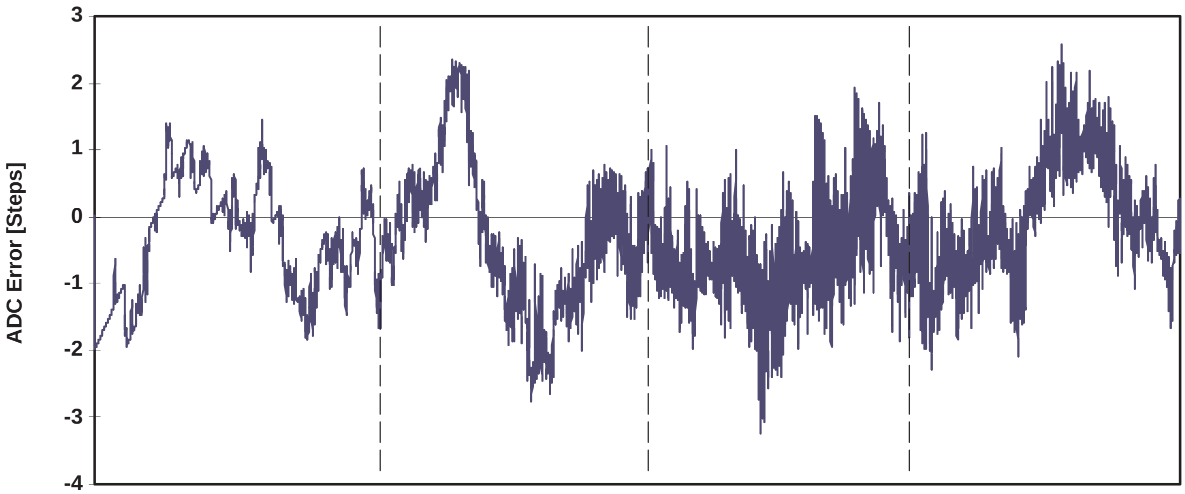


Figure 3. Error Correction With Border Fit (single linear equation)

Advantages: Only five measurements are necessary
No gaps; the monotonicity of the ADC characteristic remains
Low memory needs: 8 bytes only (four slopes and four offsets)
Good improvement of the ADC characteristic despite low expense

Disadvantages: Multiplication is necessary

The software part after each ADC measurement is as follows. For lower accuracy needs the algorithm may be simplified by the use of less accurate slopes and offsets (fewer fraction bits).

A more detailed description for the 8-bit multiplication is given in *Nonlinear Improvement of the MSP430 14-Bit ADC Characteristic*.[4] The numbers at the right margin indicate the used number format (integer.fraction)

```
; Error correction with a single equation per range
; 8–bit arithmetic. Cycles needed:
; Subdivision = 0:    51 cycles
; Subdivision > 3Fh: 100 cycles
;
;                                                         int.frct
        MOV    &ADAT,R5            ; ADC result Ni to R5        14.0
        MOV    R5,R6               ; Address info for correction 14.0
```

```
        AND    #0FFFh,R5          ; Delete range bits             12.0
        RLA    R5                 ; Calculate subdivision         13.0
        RLA    R5                 ; Prepare (Ni/4096–n)x128       14.0
        RLA    R5                 ; 7 bit ADC info to high byte   15.0
        SWPB   R5                 ; ADC info to low byte 0...7Fh  7.0
        MOV.B  R5,IROP1           ; To MPY operand register       7.0
        SWPB   R6                 ; MSBs to low byte 0...3Fh      6.0
        RRA.B  R6                 ; Calculate coeff. address      5.0
        RRA.B  R6                                                 4.0
        RRA.B  R6                 ; 2n (Range) in R6  0...07h     3.0
        BIC    #1,R6              ; 0...06h: address of slope a1  3.0
        MOV.B  TAB1(R6),IROP2L    ; Slope a1                      0.9
        CALL   #MPYS8             ; (Ni/4096–n)x 128 x a1         ±5.9
        RLA    IRACL              ; Slope part to a0 format       ±5.10
        SWPB   IRACL              ;                               ±5.2
        SXT    IRACL              ; To 16–bit format              ±5.2
        MOV.B  TAB0(R6),R5        ; Offset a0                     ±5.2
        SXT    R5                 ; To 16–bit format
        ADD    R5,IRACL           ; Ni + correction               ±5.2
        RRA    IRACL              ;                               ±5.1
        RRA    IRACL              ; Carry is used for rounding    ±5.0
        ADDC   &ADAT,IRACL        ; Corrected result Nicorr       14.0
        ...                       ; Use Nicorr in IRACL
;
; The 8 RAM bytes starting at label TAB1 contain the correction
; info a1 and a0. The bytes are loaded during the calibration
;
        .bss   TAB1,1             ; Range A a1: lin. coefficient   ±0.9
        .bss   TAB0,1             ; a0: constant coefficient       ±5.2
        .bss   TABx,6             ; Ranges B, C, D: a1, a0. (like above)
; Run time optimized 8–bit Multiplication Subroutines
;
IROP1  .EQU   R14                ; Unsigned ADC result (7Fh max.)
IROP2L .EQU   R13                ; Signed factor (80h...7Fh)
IRACL  .EQU   R12                ; Signed result word
;
; Signed multiply subroutine: IROP1 x IROP2L –> IRACL
;
MPYS8  CLR IRACL                 ; 0 –> 16 bit RESULT
       TST.B  IROP2L             ; Sign of factor (slope a1)
       JGE    MACU8              ; Positive sign: proceed
       SWPB   IROP1              ; Negative
       SUB    IROP1,IRACL        ; Correct result
       SWPB   IROP1
```

```
;
MACU8  BIT.B  #1,IROP1            ; Test actual bit (LSB)
       JZ     L$01                ; If 0: do nothing
       ADD    IROP2L,IRACL        ; If 1: add multiplier to result
L$01   RLA    IROP2L              ; Double multiplier IROP2
       RRC.B  IROP1               ; Next bit of IROP1 to LSB
       JNZ    MACU8               ; If IROP1 = 0: finished
       RET
```

EXAMPLE: The ADC is measured at the five borders of the ADC ranges. The measured errors—device 1 is used—are shown below. The correction coefficients for the range C are calculated. The correction coefficients for the other three ranges may be calculated the same way, using the appropriate border errors.

ADC Step	50	4096	8192	12288	16330
Error [Steps]	–6	–13	–10	0	–3

Error coefficients for the range C:

$$a1 = -\frac{eu - el}{128} = -\frac{0-(-10)}{128} = -\frac{10}{128} = -0.078125$$

$$a0 = -el = -(-10) = +10$$

Correction: $$\left(\frac{Ni}{4096} - n\right) \times 128 \times a1 + a0 = \left(\frac{Ni}{4096} - 2\right) \times 128 \times (-0.078125) + 10.0$$

The correction for the ADC step 11000—located in range C—is calculated:

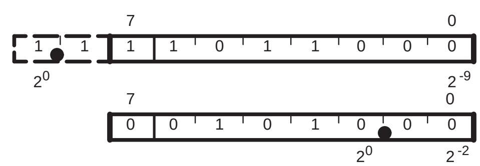
$$\left(\frac{11000}{4096} - 2\right) \times 128 \times (-0.078125) + 10.0 = +3.1$$

Corrected ADC sample: $Ni_{corr} = Ni + 3.1$

Valid for the ADC step 11000

Format: a1: ±0.9 $-0.078125/2^{-9} = -40 = D8h$

a0: ±5.2 $+10.0/2^{-2} = +40 = 28h$



The number of fractional bits for a1 is derived from the following consideration: a1 is maximally ±0.15625 (see Table 1). This value must be possible with the largest number that can be expressed with a signed 8-bit number (7Fh):

$$7Fh \times 2^{-9} > 0.15625 = \frac{20}{128} > 7Fh \times 2^{-10}$$

$$0.24805 > 0.15625 = \frac{20}{128} > 0.124025$$

This leads to a valency of 2^{-9} for the LSB of the 8-bit number. The detailed explanation for the calculation of the correction coefficients is given in *Nonlinear Improvement of the MSP430 14-Bit ADC Characteristic*,[4] *Calculation of the 8-bit Numbers*.

1.2.1.2 Multiple Linear Equations per Range

The ADC is measured at (p+1) equally distributed points over the full ADC range ($p = 2^m \geq 8$). These (p+1) results are used for the calculation of the offset and the slope of p linear equations valid for the p sections. The formula for the offset a0 and the slope a1 for each of the p linear equations is (8-bit arithmetic):

$$Ni_{corr} = Ni + \left[\left(\frac{Ni \times p}{2^{14}} - n_1 \right) \times 128 \times a_1 + a_0 \right]$$

$$a_1 = -\frac{(e_u - e_l)}{128} \qquad a_0 = -e_l$$

Where:

Ni_{corr}	= Corrected ADC sample	[Steps}
Ni	= Measured ADC sample (noncorrected)	[Steps]
n_1	= Value of the MSBs of Ni (0 to p–1)	
p	= Number of sections over the full ADC range (8 for Figure 4)	
a_1	= Slope of the correction	
a_0	= Offset of the correction	[Steps]
e_u	= Error of the ADC at the upper border of the section	[Steps]
e_l	= Error of the ADC at the lower border of the section	[Steps]

n1 ranges from 0 to (p–1) and has a length of $log_2\ p$ bits. This means for n1:

- Two linear equations per range (Figure 4): value is 0...7, length is $log_2\ 8 = 3$ bits;
- Four linear equations per range (Figure 6): value is 0...15, length is $log_2\ 16 = 4$ bits;

The term $\left(\frac{Ni \times p}{2^{14}} - n_1 \right) \times 128$ in the equation above is the adaptation of a complete section—here a half range—to 128 subdivisions. The calculation is made by simple shifts and logical AND instructions

1.2.1.3 Two Linear Equations per Range

The principle for two linear equations per range (p=8) is shown in Figure 4.

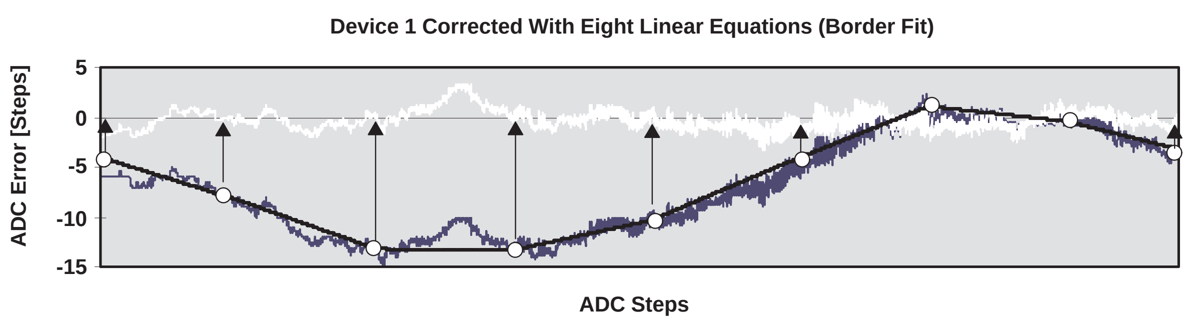


Figure 4. Principle of the Correction With Border Fit (two linear equations per range)

The statistical results for two linear equations per range are:

	Full range	Ranges A and B only
Mean Value:	–0.29 Steps	–0.02 Steps
Range:	6.49 Steps	5.3 Steps
Standard Deviation:	0.97 Steps	1.06 Steps
Variance:	0.94 Steps	1.12 Steps

Figure 5 shows the result in a graph.

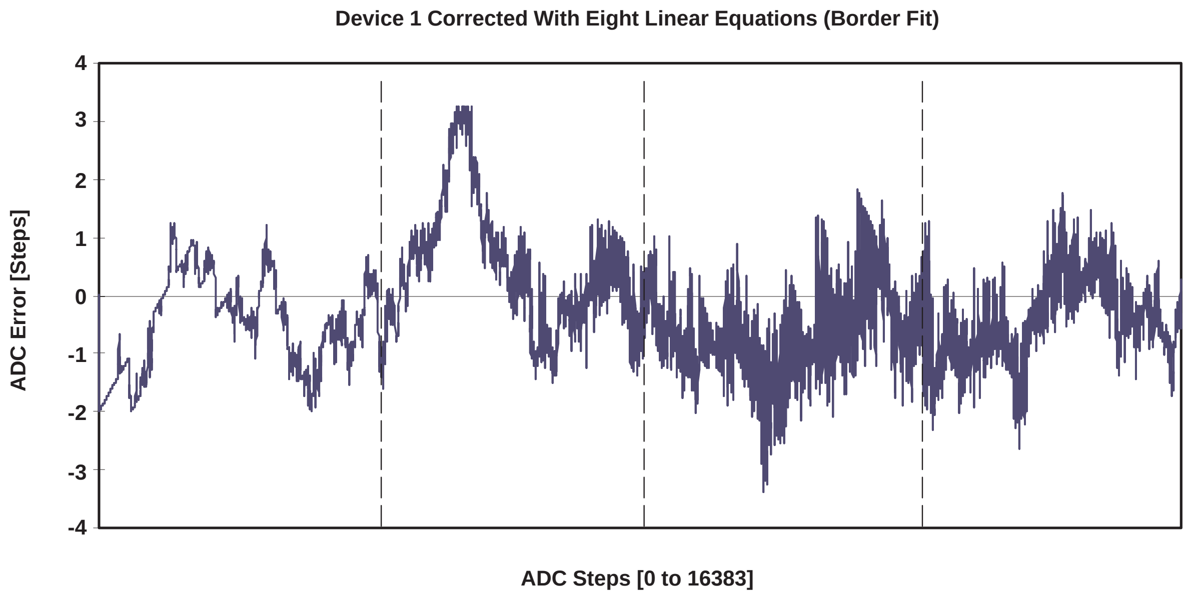


Figure 5. Error Correction With Border Fit (two linear equations per range)

Advantages: Only few measurements are necessary (p+1). Nine for the example above
No gaps; the monotonicity of the ADC characteristic is preserved
Better correction than with a single linear equation per range
Low memory needs: $2 \times p$ bytes (16 for the example)

Disadvantages: Multiplication is necessary

The software is the same as shown in section 1.2.2.2, *Multiple Linear Equations per Range*.

EXAMPLE: The ADC is corrected with eight sections, each one with a length of 2048 steps (p = 8). The measured errors—(device 1 of *Architecture and Function of the MSP430 14-Bit ADC*, SLAA045 is used)—are shown below. The correction coefficients for the lower section of range C—ADC steps 8192 to 10240 (n1 = 4)—are calculated. The correction coefficients for the other seven sections are calculated the same way.

ADC Step	50	2048	4096	6144	8192	10240	12288	14336	16330
n1	0	1	2	3	4	5	6	7	7
Error [Steps]	–6	–8	–13	–13	–10	–5	0	0	–3

Error coefficients for the lower section of range C:

$$a_1 = -\frac{e_u - e_l}{128} = -\frac{-5-(-10)}{128} = -\frac{5}{128} = -0.0390625$$

$$a_0 = -e_l = -(-10) = +10$$

Correction: $$\left(\frac{Ni \times p}{2^{14}} - n1\right) \times 128 \times a_1 + a_0 = \left(\frac{Ni \times 8}{2^{14}} - 4\right) \times 128 \times (-0.03906) + 10.0$$

Lower section of range C

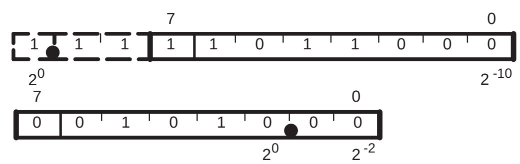
The correction for the ADC step 9000—located in the lower section of range C—is calculated (p = 8, n_1 = 4):

$$\left(\frac{9000 \times 8}{2^{14}} - 4\right) \times 128 \times (-0.0390625) + 10.0 = 50.5 \times (-0.0390625) + 10.0 = +8.027$$

Corrected ADC sample: $Ni_{corr} = Ni + 8.03$ Valid for the ADC step 9000 (range C)

Format: a_1: ±0.10 –0.039625/2^{-10} = –40 = D8h

a_0: ±5.2 +10.0/2^{-2} = 40 = 28h



1.2.1.4 Four Linear Equations per Range

The principle for four linear equations per range (p=16) is shown in Figure 6.

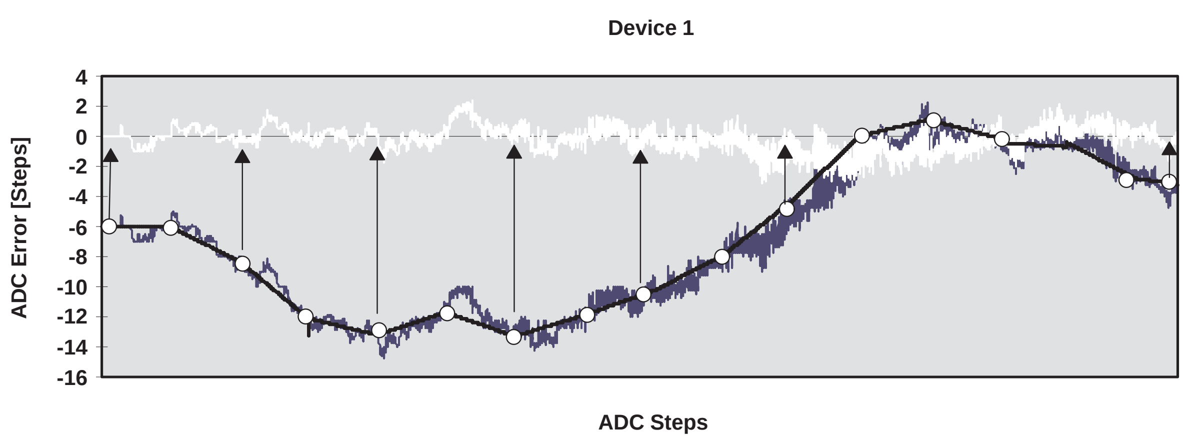


Figure 6. Principle of the Correction With Border Fit (four linear equations per range)

The statistical results for four linear equations per range are:

	Full range	Ranges A and B only
Mean Value:	–0.22 Steps	0.07 Steps
Range:	5.36 Steps	4.07 Steps
Standard Deviation:	0.83 Steps	0.65 Steps
Variance:	0.69 Steps	0.42 Steps

Figure 7 shows the result in a graph.

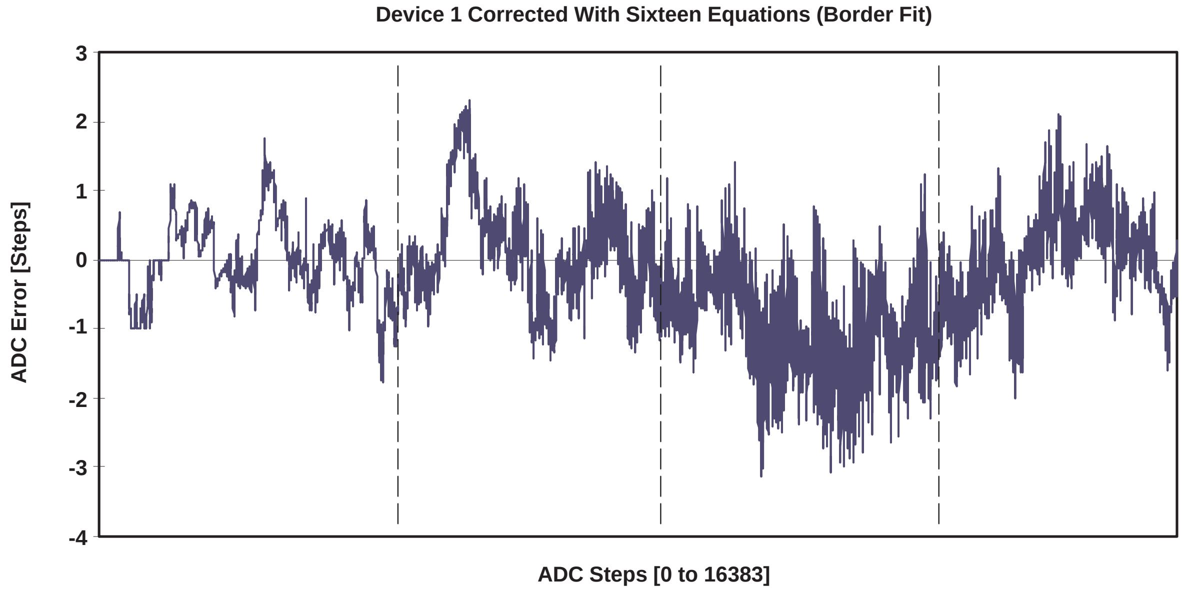


Figure 7. Error Correction With Border Fit (four linear equations per range)

Advantages: Only a few measurements are necessary (p+1). Seventeen for the example above
No gaps; the monotonicity of the ADC characteristic is preserved
Better correction than with one or two linear equations per range
Low memory requirements if p is small: $2 \times p$ bytes (32 for above example)

Disadvantages: Multiplication is necessary

For 16 linear equations for the full ADC range, the software for each ADC measurement is as follows:

```
; Error correction with four linear equations per range
; (16 for the full ADC range) 8–bit arithmetic. Cycles needed:
; Subdivision = 0:    49 cycles
; Subdivision > 3Fh: 101 cycles
;
    MOV    &ADAT,R5          ; ADC result Ni to R5           4.0
    MOV    R5,R6             ; Address info for correction
    AND    #03FFh,R5         ; Delete 4 MSBs (n1 bits)       10.0
    RLA    R5                ; Calculate subdivision         11.0
    RLA    R5                ;                               12.0
    RLA    R5                ; Prepare                       13.0
    RLA    R5                ; ((Ni x p/2^14)–n1)x 128       14.0
    RLA    R5                ; 7 bit ADC info to high byte   15.0
    SWPB   R5                ; ADC info to low byte 0...7Fh  7.0
    MOV.B  R5,IROP1          ; To MPY operand register       7.0
```

```
    SWPB   R6                    ; Calculate coeff. address       6.0
    RRA.B  R6                    ; 2 x n1 in R6  0...01Fh         5.0
    BIC    #1,R6                 ; 0...01Eh: address of slope a1  5.0
    MOV.B  TAB1(R6),IROP2L       ; Slope a1                       ±0.11
    CALL   #MPYS8                ; ((Ni x p/2^14)–n1)x 128 x a1   ±3.11
    RRA    IRACL                 ; MPY result to a0 format        ±3.10
    SWPB   IRACL                 ;                                ±3.2
    ADD.B  TAB0(R6),IRACL        ; Offset a0                      ±5.2
    SXT    IRACL                 ;                                ±5.2
    RRA    IRACL                 ;                                ±5.1
    RRA    IRACL                 ; Carry is used for rounding     ±5.0
    ADDC   &ADAT,IRACL           ; Corrected result Nicorr        14.0
    ...                          ; Proceed with Nicorr in IRACL
;
; The 32 RAM bytes starting at label TAB1 contain the corr.
; coefficients a1 and a0. The bytes are loaded during the
; initialization. 8–bit, signed numbers
;
    .bss   TAB1,1                ; Range A lowest quarter:  a1  ±0.11
    .bss   TAB0,1                ;                          a0  ±5.2
    .bss   TABx,30               ; Ranges A (3), B, C, D: a1, a0.
```

EXAMPLE: The ADC is corrected with sixteen sections, each one with a length of 1024 steps (p = 16). The measured errors (device 1 of *Architeture and Function of the MSP430 14-Bit ADC*, SLAA045 is used.) are shown below. The correction coefficients for the lowest section of range C—ADC steps 8192 to 9216 (n1 = 8)—are calculated. The correction coefficients for the other seven sections are calculated the same way.

ADC Step	8192	9216	10240
n1	8	9	10
Error [Steps]	–10	–7	–5

Error coefficients for the lower section of range C:

$$a1 = -\frac{eu - el}{128} = -\frac{-7-(-10)}{128} = -\frac{3}{128} = -0.0234375$$

$$a0 = -el = -(-10) = +10$$

Correction: $$\left(\frac{Ni \times p}{2^{14}} - n1\right) \times 128 \times a0 + a1 = \left(\frac{Ni \times 16}{2^{14}} - 8\right) \times 128 \times (-0.0234375) + 10.0$$

Lower section of range C

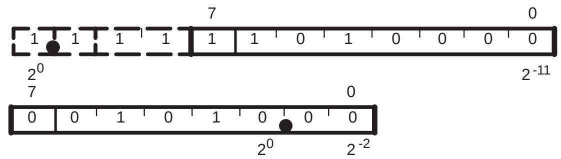
The correction for the ADC step 9000—located in the lower section of range C—is calculated ($p = 16$, $n_1 = 8$):

$$\left(\frac{9000 \times 16}{2^{14}} - 8\right) \times 128 \times (-0.0234375) + 10.0 = 101.0 \times (-0.0234375) + 10.0 = +7.63$$

Corrected ADC sample: $Ni_{corr} = Ni + 7.63$ Valid for the ADC step 9000 (range C)

Format: a_1: ±0.11 –0.0234375/2^{-11} = –48 – D0h

a_0: ±5.2 +10.0/2^{-2} = 40 = 28h



1.2.2 Linear Equations With Linear Regression

With linear regression the linear equations that best fit the measured ADC characteristic are used. This leads to good results within the ranges but may produce gaps at the borders.

The linear regression formulas (*Least Squares Method*) for the correction coefficients a1 (slope) and a0 (offset) are given below. To simplify the real time calculations, the negative values of the coefficients are used. The reasons for this are the same ones as described in section 1.1.

$$a_1 = -\frac{\dfrac{\sum_{i=1}^{i=k} N \times \sum_{i=1}^{i=k} ei}{k} - \sum_{i=1}^{i=k} N \times ei}{\dfrac{\left(\sum_{i=1}^{i=k} N\right)^2}{k} - \sum_{i=1}^{i=k} N^2} \qquad a_0 = -\left(\overline{e} - a_1 \times \overline{N}\right)$$

The mean values of N and e are defined as:

$$\overline{N} = \frac{\sum_{i=1}^{i=k} N}{k} \qquad \overline{e} = \frac{\sum_{i=1}^{i=k} ei}{k}$$

Where:

N	=	Measured ADC sample (noncorrected)	[Steps]
ei	=	Error of the ADC sample i	[Steps]
a_1	=	Slope of the correction (negated)	
a_0	=	Offset of the correction (negated)	[Steps]
k	=	Number of the measured samples	
i	=	Sample index running from 1 to k	

The value N represents different values depending on the calculation method:

- 8-bit arithmetic: the subdivision of Ni within the appropriate section. The range for N is 0...127. See the explanation given in section 1.2.1.1
- Floating Point and 16-bit arithmetic: N equals the full 14-bit ADC value Ni

The examples used are simplified due to the amount of data involved.

1.2.2.1 Single Linear Equation per Range

The ADC is measured at k points inside each of the four ranges. Out of these (4×k) results, four linear equations are calculated using the *Least Squares Method* (see above formulas). The four slopes and offsets are stored in the RAM or in EEPROM. The formula for the corrected value Nicorr is:

$$Ni_{corr} = Ni + \left[\left(\frac{Ni}{4096} - n\right) \times 128 \times a1 + a0\right]$$

Where:

n = Range number (0...3) for ADC ranges A...D)
a1 = Slope calculated by the host or MSP430
a0 = Offset calculated by the host or MSP430 [Steps]
k = Number of samples for each linear equation (range)

The term $\left(\frac{Ni}{4096} - n\right) \times 128$ of the above equation is the adaptation of a complete section—here a full range—to 128 subdivisions. The calculation is made by simple shifts and logical AND instructions. See the initialization part of the example below.

The principle of this method is shown in Figure 8, the eight measured samples are drawn only in range A (k = 8):

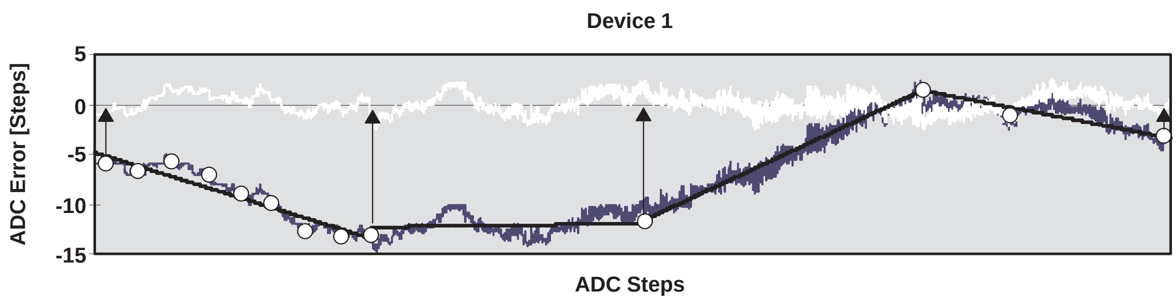


Figure 8. Principle of the Linear Regression Method (single linear equation per range)

	Full range	Ranges A and B only
Mean Value:	0.03 Steps	0.12 Steps
Range:	5.09 Steps	4.85 Steps
Standard Deviation:	0.94 Steps	1.00 Steps
Variance:	0.88 Steps	1.00 Steps

The statistical results for 8 and 16 measurements per range are shown below: as it can be seen, 16 samples per range improve the final result only marginally.

	8 Samples per range	16 Samples per Range
Mean Value:	0.03 Steps	0.07 Steps
Range:	5.09 Steps	5.04 Steps
Standard Deviation:	0.94 Steps	0.92 Steps
Variance:	0.88 Steps	0.85 Steps

Figure 9 shows the result of this method: eight samples per range are measured (k=8). Note the small range of only ±3 steps.

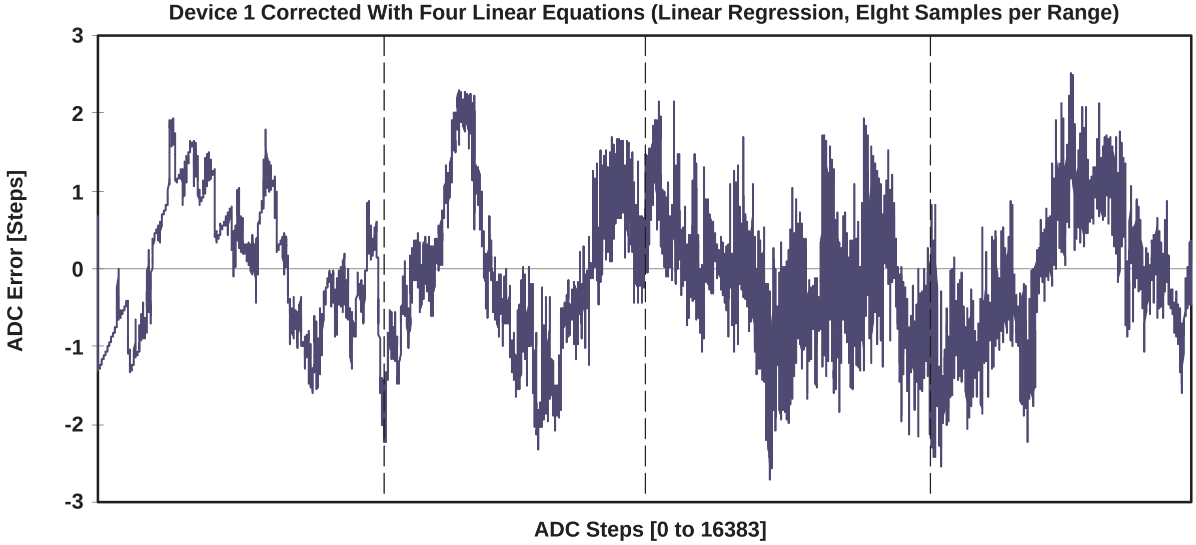


Figure 9. Error Correction With Linear Regression (single linear equation per range)

Advantages: Good adaptation to the ADC characteristic

Disadvantages: One multiplication is necessary
Small gaps at the borders of the four ranges
Calculation of the linear regression is necessary during the calibration

Device 1 does not show gaps at the borders of the four ranges—which is purely random—therefore another device that shows this disadvantage of the method more clearly is included in Figure 10. Note the gaps between the ranges A and B and the ranges B and C.

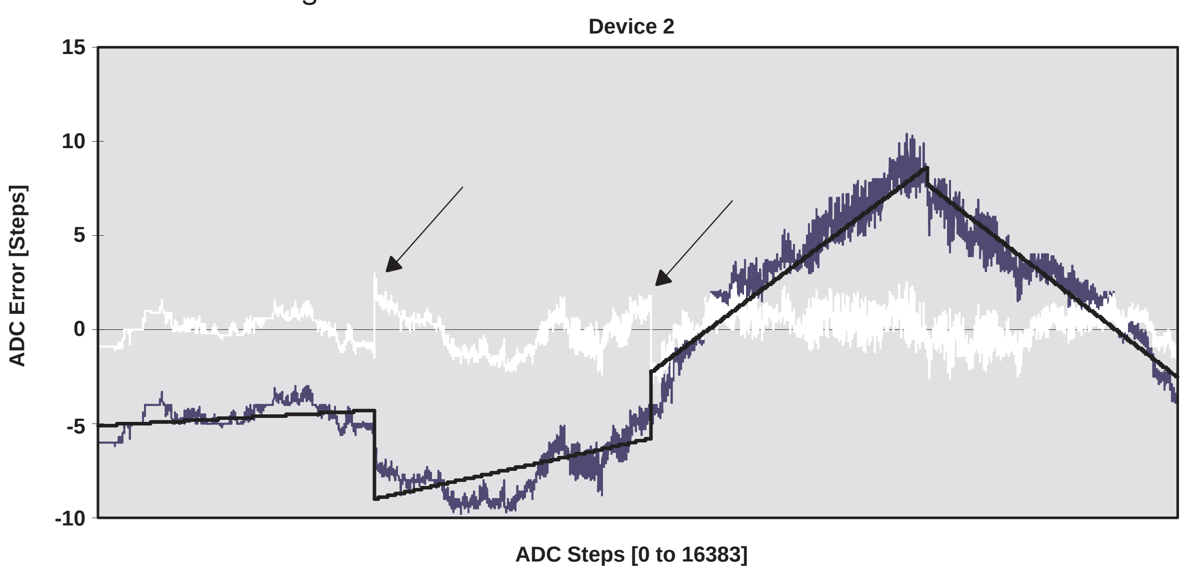


Figure 10. Device 2[1] Showing the Typical Gaps at the Range Borders

[1] This is device 2 from *Archecture and Function of the MSP430 14-Bit ADC Application Report #SLAA045.*

The correction software for the 8-bit arithmetic is identical to the one shown in section *Single Linear Equation per Range (Border Fit)*, section 1.2.1.1.

Here an additional solution with 16-bit integer arithmetic is given.

```
; Error correction with a single equation per range
; 16-bit arithmetic. Cycles needed:
; ADAT value = 0000h:  47 cycles
; ADAT value = 3FFFh: 178 cycles
;
      MOV    &ADAT,IROP1      ; ADC result Ni to MPY reg.      14.0
      MOV    IROP1,R6         ; Calculation of coeff. address  14.0
      SWPB   R6               ; MSBs to low byte 0...3Fh       6.0
      RRA.B  R6               ;                                5.0
      RRA.B  R6               ; 4n (Range) in R6  0...0Fh      4.0
      BIC    #3,R6            ; 0...0Ch: address of slope a1   4.0
      MOV    TAB1(R6),IROP2L  ;  Slope a1                      0.22
      CALL   #MPYS            ; Ni x a1                        ±4.22
      RRA    IRACM            ; Only HI result is used         ±4.5
      RRA    IRACM            ; To format 4.3 of offset a0     ±4.4
      RRA    IRACM            ;                                ±4.3
      ADD    TAB0(R6),IRACM   ; Add Offset a0                  ±5.3
      RRA    IRACM            ; Nicorr = Ni x a1 + a0          ±5.2
      RRA    IRACM            ;                                ±5.1
      RRA    IRACM            ; Carry is used for rounding     ±5.0
      ADDC   &ADAT,IRACM      ; Nicorr in IRACM                14.0
      ...                     ; Proceed with corr. result Nicorr
;
; The 16 RAM bytes starting at label TAB1 contain the
; correction info a1 and a0 for all four ranges. The bytes
; are loaded during the calibration
;
      .bss   TAB1,2           ; Range A a1: lin. coefficient   ±0.22
      .bss   TAB0,2           ; a0: constant coefficient       ±5.3
      .bss   TABx,12          ; Ranges B, C, D: a1, a0.
; Run time optimized 16–bit Multiplication Subroutines
;
IROP1  .EQU   R11             ; Unsigned ADC result (0...3FFFh)
IROP2L .EQU   R12             ; Signed factor (8000h...7FFFh)
IROP2M .EQU   R13             ; High word of signed factor (0)
IRACL  .EQU   R14             ; Result word low
IRACM  .EQU   R15             ; Result word high
;
; Signed multiply subroutine: IROP1 x IROP2L –> IRACM|IRACL
;
```

```
MPYS   CLR    IRACL            ; 0 –> result word low
       CLR    IRACM            ; 0 –> result word high
       TST    IROP2L           ; Sign of factor a1
       JGE    MACU             ; Positive sign: proceed
       SUB    IROP1,IRACM      ; Correct result
MACU   CLR    IROP2M           ; Clear MSBs multiplier
L$002  BIT    #1,IROP1         ; Test actual bit (LSB)
       JZ     L$01             ; If 0: do nothing
       ADD    IROP2L,IRACL     ; If 1: add multiplier to result
       ADDC   IROP2M,IRACM
L$01   RLA    IROP2L           ; Double multiplier IROP2
       RLC    IROP2M           ;
;
       RRC    IROP1            ; Next bit of IROP1 to LSB
       JNZ    L$002            ; If IROP1 = 0: finished
       RET
```

EXAMPLE: (8-bit arithmetic). The ADC is measured at five points of the ADC range A (n = 0). The measured errors—device 1 is used—are shown below. The correction coefficients for the range A are calculated with the linear regression method. The correction coefficients for the other three ranges may be calculated the same way. The used numbers are shaded.

ADC Step	50	1024	2048	3072	4096
Subdivision	1.56	32	64	96	128
Error [Steps]	–6	–6	–8	–12	–13

The correction coefficients for the range A (n=0), are calculated with the formulas shown in section 1.2.2.

$$a_1 = +0.06326 \qquad \textit{Negated result of linear coefficient}$$

$$a_0 = +4.9312 \qquad \textit{Negated result of constant coefficient}$$

$$\text{Correction:} \quad \left[\left(\frac{Ni}{4096} - 0\right) \times 128 \times a_1 + a_0\right] = \left(\frac{Ni}{32} \times 0.06326 + 4.9312\right)$$

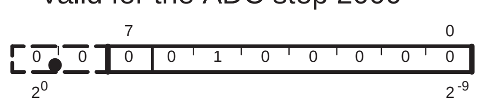
The correction for the ADC step 2000—located in range A—is calculated:

$$\frac{Ni}{32} \times 0.06326 + 4.93 = \frac{2000}{32} \times 0.06326 + 4.93 = +8.88$$

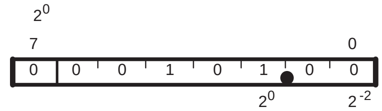
Corrected ADC sample: $Ni_{corr} = Ni + 8.9$

Format: Valid for the ADC step 2000

a_1: ±0.9 $+0.06326/2^{-9} = +32.4 \approx 20h$



a_0: ±5.2 $+4.93/2^{-2} = +19.7 \approx 14h$



EXAMPLE: (16-bit arithmetic). The ADC is measured at five points of the ADC range C. The measured errors—device 1 is used—are shown in the table below. The correction coefficients for the range C are calculated with the linear regression method. The correction coefficients for the other three ranges may be calculated the same way. The used numbers are shaded.

ADC Step	8192	9216	10240	11254	12288
Error [Steps]	–9.6	–8.6	–5.2	–1	+0.1

The correction coefficients for the range C are calculated with the formulas shown in section 1.2.2. The full 14-bit ADC result is used for the calculations due to the available 16 bits of resolution.

$a_1 = -0.0026381701$ *Negated result of linear coefficient*

$a_0 = +31.8695$ *Negated result of constant coefficient*

Correction: $Ni \times a_1 + a_0 = Ni \times (-0.00263817) + 31.8695$

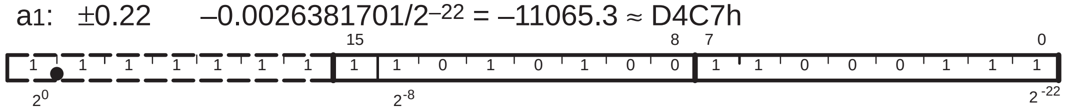
The correction for the ADC step 12000—located in range C—is calculated:

$$Ni \times (-0.00263817) + 31.8695 = 12000 \times (-0.00263817) + 31.8695 = +0.204$$

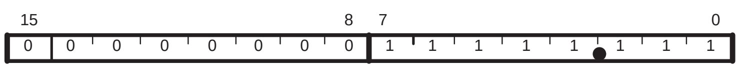
Corrected ADC sample: $Ni_{corr} = Ni + 0.2$ Valid for the ADC step 12000

Format:

a1: ±0.22 –0.0026381701/2^{-22} = –11065.3 ≈ D4C7h



a0: ±5.3 +31.86958564/2^{-3} = +254.96 ≈ 00FFh



1.2.2.2 Multiple Linear Equations per Range

The ADC is measured at (p × k) points over the four ranges (p = $2^m \geq 8$). Out of these (p × k) results p linear equations are calculated using the Least Squares Method. The calculated slopes and offsets are stored in the RAM or in EEPROM. The formula for the correction is:

$$Ni_{corr} = Ni + \left[\left(\frac{Ni \times p}{2^{14}} - n_1 \right) \times 128 \times a_1 + a_0 \right]$$

Where:

Ni_{corr}	= Corrected ADC sample	[Steps]
Ni	= Measured ADC sample (noncorrected)	[Steps]
p	= Number of sections for the full ADC range. p is a power of 2.	
n_1	= Value of the MSBS of Ni. n1 ranges from 0 to (p–1)	
a_1	= Slope of the correction	
a_0	= Offset of the correction	
k	= Number of samples for each linear equation (section)	

The value n_1 is explained in section *Multiple Linear Equations (Border Fit)*, section 1.2.1.2.

The term $\left(\frac{Ni \times p}{2^{14}} - n1\right) \times 128$ in the above equation is the adaptation of a complete section—here a half range—to 128 subdivisions. The calculation is made by simple shifts and logical AND instructions.

The principle of this method—with four samples within each one of the eight sections (k = 4, p = 8)—is shown in Figure 11, the ADC samples are shown only in range A:

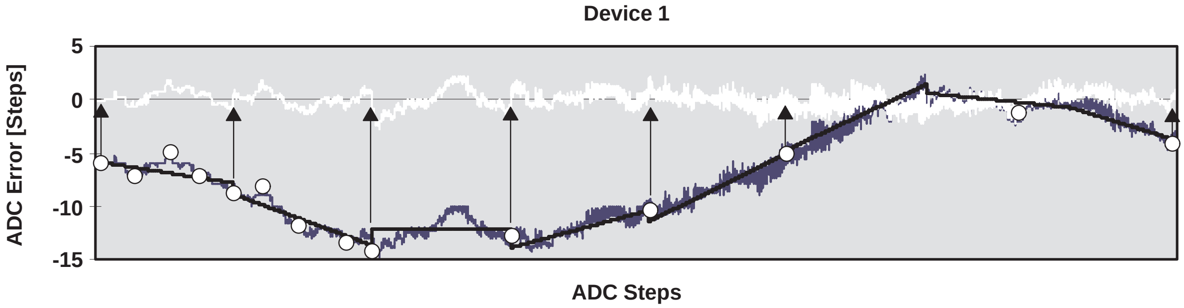


Figure 11. Principle of the Linear Regression Method (two linear equations per range)

The statistical results for 16 points per range—eight samples for each one of the eight linear equations (k = 8, p = 8)—are:

	Full range	Ranges A and B only
Mean Value:	–0.03 Steps	+0.09 Steps
Range:	4.84 Steps	4.80 Steps
Standard Deviation:	0.78 Steps	0.79 Steps
Variance:	0.61 Steps	0.63 Steps

The result is shown in Figure 12. Note the error range of this figure: only ±3 ADC steps.

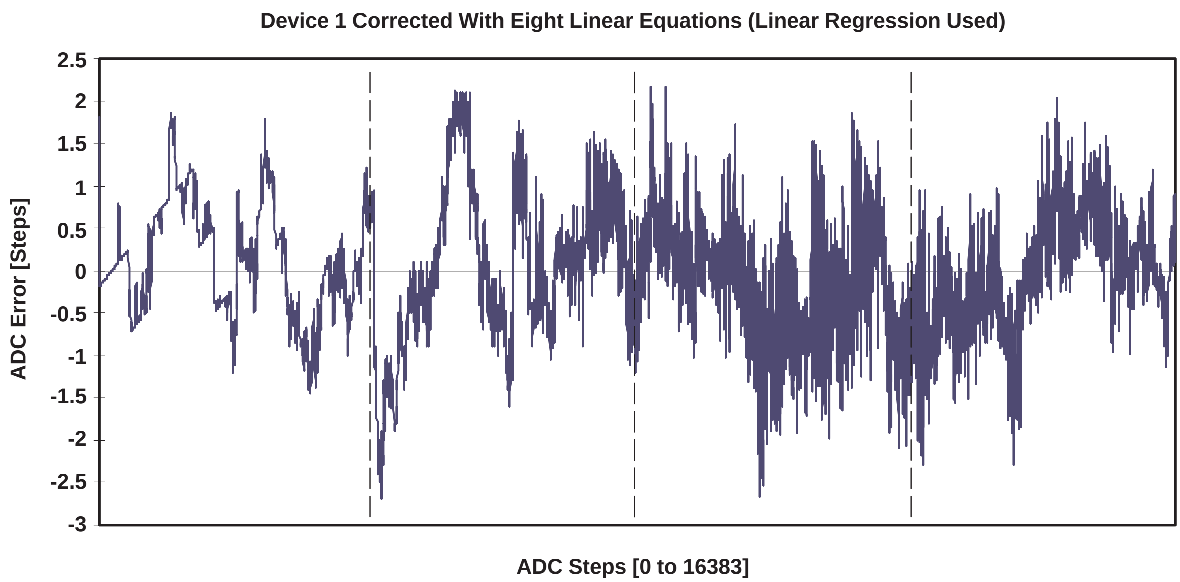


Figure 12. Error Correction With Linear Regression (two linear equations per range)

Advantages: Very good adaptation to the ADC characteristic
Method can be adapted to specific needs with more equations per range

Disadvantages: Multiplication is necessary
Small gaps at the borders of the four ranges
Calculation of linear regression is necessary during calibration
Many measurements are necessary during calibration (64 with the above example)

The correction software for the 8-bit arithmetic:

```
; Error correction with two linear equations per range
; (8 for the full ADC range) 8–bit arithmetic. Cycles needed:
; Subdivision = 0:    48 cycles
; Subdivision > 3Fh:  97 cycles
;
   MOV    &ADAT,R5         ; ADC result Ni to R5            14.0
   MOV    R5,R6            ; Address info for correction    14.0
   AND    #07FFh,R5        ; Delete 3 MSBs (n1 bits)        11.0
   RLA    R5               ; Calculate subdivision
   RLA    R5               ; Prepare                        13.0
   RLA    R5               ; ((Ni x p/2^14)–n1)x 128        14.0
   RLA    R5               ; 7 bit ADC info to high byte    15.0
   SWPB   R5               ; ADC info to low byte 0...7Fh   7.0
   MOV.B  R5,IROP1         ; To MPY operand register        7.0
;
   SWPB   R6               ; Calculate coeff. address       6.0
   RRA.B  R6               ; 0...3Fh to 0...1Fh             5.0
   RRA.B  R6               ; 2 x n1 in R6  0...0Fh          4.0
   BIC    #1,R6            ; 0...0Eh: address of slope a1   4.0
   MOV.B  TAB1(R6),IROP2L  ; Slope a1   ±0.10
   CALL   #MPYS8           ; ((Ni x p/2^14)–n1)x 128 x a1   ±4.10
   SWPB   IRACL            ; MPY result to a0 format  ±     4.2
   ADD.B  TAB0(R6),IRACL   ; (nnn)x 128 x a1 + a0           ±5.2
   SXT    IRACL            ;                                ±5.2
   RRA    IRACL            ; To integer format              ±5.1
   RRA    IRACL            ; Carry is used for rounding     ±5.0
   ADDC   &ADAT,IRACL      ; Corrected result Nicorr        14.0
   ...                     ; Proceed with Nicorr in IRACL
;
; The 16 RAM bytes starting at label TAB contain the correction
; coefficients a1 and a0. The bytes are loaded during the
; initialization. 8-bit, signed numbers
;
   .bss   TAB1,1           ; Range A: a1                    ±0.9
```

```
.bss   TAB0,1          ; a0                              ±5.2
.bss   TABx,14         ; Range B, C, D: a1, a0.
```

EXAMPLE: The ADC ranges are split into two sections each. The measured errors of five points located in the upper section of range B—device 1 is used—are shown below (k = 4, p = 8). The correction coefficients for this section are calculated with the linear regression method. The correction coefficients for the other seven sections may be calculated the same way.

ADC Step	6144	6656	7168	7680	8192
Subdivision	0	32	64	96	128
Error [Steps]	–14	–13.6	–12	–10.5	–9.6

The correction coefficients a_1 and a_0 for the upper section of range B (n_1 = 3) are calculated with the formulas shown in section 1.2.2. The subdivision of the ADC step (0 to 127) is used (8-bit arithmetic).

$$a_1 = +0.03719 \qquad \textit{Negated value}$$
$$a_0 = +14.32 \qquad \textit{Negated value}$$

Correction:

$$\left[\left(\frac{Ni \times p}{2^{14}} - n_1\right) \times 128 \times a_0 + a_1\right] = \left[\left(\frac{Ni \times 8}{2^{14}} - 3\right) \times 128 \times (-0.03719) + 14.32\right]$$

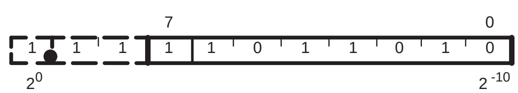
The correction for the ADC step 7000—located in the upper section of range B—is calculated:

$$\left(\frac{7000 \times 8}{2^{14}} - 3\right) \times 128 \times (-0.03719) + 14.32 = +12.33$$

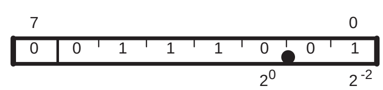
Corrected ADC sample: $Ni_{corr} = Ni + 12.3$ Valid for ADC step 7000 range B

Format:

a_1: ±0.10 –0.03719/2^{-10} ≈ –38 = DAh



a_0: ±5.2 +14.32/2^{-2} = 57.3 ≈ 39h



2 Additional Information

The application report *Nonlinear Improvement of the MSP430 14-Bit ADC Characteristic*[4] shows nonlinear methods such as quadratic and cubic corrections for the improvement of the 14-bit analog-to-digital converter of the MSP430. Also included are the integer multiplication subroutines for the fast correction software and considerations to the obtainable accuracy with the 8-bit software. Finally all explained correction methods presented are compared by ROM and RAM needs, accuracy improvement, and required CPU cycles.

3 References

1. *Architecture and Function of the MSP430 14-Bit ADC Application Report*, 1999, Literature #SLAA045
2. *Application Basics for the MSP430 14-Bit ADC Application Report*, 1999, Literature #SLAA046
3. *Additive Improvement of the MSP430 14-Bit ADC Characteristic Application Report*, 1999, Literature #SLAA047
4. *Nonlinear Improvement of the MSP430 14-Bit ADC Characteristic Application Report*, 1999, Literature #SLAA050
5. *MSP430 Application Report*, 1998, Literature #SLAAE10C
6. Data Sheet MSP430C325, MSP430P323, 1997, Literature #SLASE06B
7. *MSP430 Family Architecture Guide and Module Library*, 1996, Literature #SLAUE10B

Appendix A Definitions Used With the Application Examples

```
; HARDWARE DEFINITIONS
;
ACTL  .equ  0114h ; ADC control register: control bits
ADAT  .equ  0118h ; ADC data register (12 or 14–bits)
```