SMART ARM-based MCUs

AT14164: User Calibration of Internal Temperature Sensor - SAM R21

APPLICATION NOTE

Introduction

This application note explains about calibrating and compensating the errors of temperature measurements from the internal temperature sensor of Atmel® | SMART SAM R21.

It provides the step-by-step information to:

- Store the calibration values in the NVM user row during production calibration
- Obtain an accurate temperature reading by compensating the temperature measurements using the calibration value, stored in the NVM user row

The firmware package available with this application note contains the code implementation that supports the methods described in this application note. This code implementation can be used to obtain an accurate temperature reading.

Table of Contents

1. Internal Temperature Sensor of SAM R21

The internal temperature sensor of SAM R21 can be used to determine an absolute temperature. The temperature sensor output can be routed to an ADC input channel by setting the bit TSEN (Bit 1 – Voltage References System (VREF) Control). The ADC can be configured to measure this output voltage.

The sensor will output a highly linear voltage proportional to the temperature. Due to process variations the output of the temperature sensor might vary from one chip to another. Also, the internal voltage reference used as the ADC reference voltage varies with the temperature. To obtain an accurate temperature reading from the internal temperature sensor, these variations must be compensated.

The characteristics of the internal temperature sensor are as follows.

Table 1-1 Temperature Sensor Characteristics

Parameter	Conditions	Min.	Typ.	Max.	Units
Temperature sensor output voltage	T= 25°C, V_{DDANA} = 3.3V	-	0.667	-	V
Temperature sensor slope		2.3	2.4	2.5	mV/°C
Variation over V_{DDANA} voltage	V_{DDANA}=1.8V to 3.6V	-1.7	1	3.7	mV/V
Temperature sensor accuracy	Using the method described in “Software-based Refinement of the Actual Temperature” section of SAM R21 Datasheet.	-10	-	10	°C

The upcoming topics describe the method to compensate the errors in temperature measurements using the internal temperature sensor.

2. Compensating Internal Voltage Reference Variation

2.1. Software-based Refinement of the Actual Temperature

The temperature sensor behavior is linear but depends on several parameters such as the internal voltage reference, which itself depends on the temperature. To take this into account, each device contains a Temperature Log row with data measured and written during the production tests. These calibration values should be read by software to infer the most accurate temperature readings possible.

This Software Temperature Log row can be read at address 0x00806030

This section specifies the Temperature Log row content and explains how to refine the temperature sensor output using the values in the Temperature Log row.

2.2. Temperature Log Row

All values in this row were measured in the following conditions:

- $V_{DDIN} = V_{DDIO} = V_{DDANA} = 3.3V$
- ADC Clock speed = 1MHz
- ADC mode: Free running mode, ADC averaging mode with 4 averaged samples
- ADC voltage reference = 1.0V internal reference (INT1V)
- ADC input = Temperature sensor

Table 2-1 Temperature Log Row Content

Bit position	Name	Description
7:0	ROOM_TEMP_VAL_INT	Integer part of room temperature in °C
11:8	ROOM_TEMP_VAL_DEC	Decimal part of room temperature
19:12	HOT_TEMP_VAL_INT	Integer part of hot temperature in °C
23:20	HOT_TEMP_VAL_DEC	Decimal part of hot temperature
31:24	ROOM_INT1V_VAL	2's complement of the internal 1V reference drift at room temperature (versus a 1.0 centered value)
39:32	HOT_INT1V_VAL	2's complement of the internal 1V reference drift at hot temperature (versus a 1.0 centered value)
51:40	ROOM_ADC_VAL	12-bit ADC conversion at room temperature
63:52	HOT_ADC_VAL	12-bit ADC conversion at hot temperature

The temperature sensor values are logged during test production flow for Room and Hot insertions:

- ROOM_TEMP_VAL_INT and ROOM_TEMP_VAL_DEC contains the measured temperature at room insertion (e.g. for ROOM_TEMP_VAL_INT=25 and ROOM_TEMP_VAL_DEC=2, the measured temperature at room insertion is 25.2°C).
- HOT_TEMP_VAL_INT and HOT_TEMP_VAL_DEC contains the measured temperature at hot insertion (e.g. for HOT_TEMP_VAL_INT=83 and HOT_TEMP_VAL_DEC=3, the measured temperature at room insertion is 83.3°C).

The temperature log row also contains the corresponding 12-bit ADC conversions of both Room and Hot temperatures:

- ROOM_ADC_VAL contains the 12-bit ADC value corresponding to (ROOM_TEMP_VAL_INT, ROOM_TEMP_VAL_DEC)
- HOT_ADC_VAL contains the 12-bit ADC value corresponding to (HOT_TEMP_VAL_INT, HOT_TEMP_VAL_DEC)

The temperature log row also contains the corresponding 1V internal reference of both Room and Hot temperatures:

- ROOM_INT1V_VAL is the 2’s complement of the internal 1V reference value corresponding to (ROOM_TEMP_VAL_INT, ROOM_TEMP_VAL_DEC)
- HOT_INT1V_VAL is the 2’s complement of the internal 1V reference value corresponding to (HOT_TEMP_VAL_INT, HOT_TEMP_VAL_DEC)
- ROOM_INT1V_VAL and HOT_INT1V_VAL values are centered around 1V with a 0.001V step. In other words, the range of values [0,127] corresponds to [1V, 0.873V] and the range of values [-1, -127] corresponds to [1.001V, 1.127V]. INT1V == 1 - (VAL/1000) is valid for both ranges.

2.3. Using Linear Interpolation

For concise equations, we will use the following notations:

- (ROOM_TEMP_VAL_INT, ROOM_TEMP_VAL_DEC) is denoted temp_R
- (HOT_TEMP_VAL_INT, HOT_TEMP_VAL_DEC) is denoted temp_H
- ROOM_ADC_VAL is denoted ADC_R, its conversion to Volt is denoted V_{ADCR}
- HOT_ADC_VAL is denoted ADC_H, its conversion to Volt is denoted V_{ADCH}
- ROOM_INT1V_VAL is denoted INT1V_R
- HOT_INT1V_VAL is denoted INT1V_H

Using the (temp_R, ADC_R) and (temp_H, ADC_H) points, using a linear interpolation we have the following equation:

$$\left(\frac{V_{\text{ADC}} + -V_{\text{ADCR}}}{\text{temp} + -\text{temp}_R}\right) = \left(\frac{V_{\text{ADCH}} + -V_{\text{ADCR}}}{\text{temp}_H + -\text{temp}_R}\right)$$

Given a temperature sensor ADC conversion value ADC_m, we can infer a coarse value of the temperature temp_C as:

$$\text{temp}_C = \text{temp}_R + \left[\frac{\left\{\left(\text{ADC}_m \cdot \frac{1}{(2^{12} + -1)}\right) + -\left(\text{ADC}_R \cdot \frac{\text{INT1}V_R}{(2^{12} + -1)}\right)\right\} \cdot (\text{temp}_H + -\text{temp}_R)}{\left\{\left(\text{ADC}_H \cdot \frac{\text{INT1}V_H}{(2^{12} + -1)}\right) + -\left(\text{ADC}_R \cdot \frac{\text{INT1}V_R}{(2^{12} + -1)}\right)\right\}}\right]$$

[Equation 1]

Note:

1. In the previous expression, we have added the conversion of the ADC register value to be expressed in V.
2. This is a coarse value because we assume INT1V=1V for this ADC conversion.

Using the (temp_R, INT1V_R) and (temp_H, INT1V_H) points, using a linear interpolation we have the following equation:

$$\left(\frac{\text{INT1}V + -\text{INT1}V_R}{\text{temp} + -\text{temp}_R}\right) = \left(\frac{\text{INT1}V_H + -\text{INT1}V_R}{\text{temp}_H + -\text{temp}_R}\right)$$

Then using the coarse temperature value, we can infer a closer to reality INT1V value during the ADC conversion as:

$$\mathrm{INT1}V_m = \mathrm{INT1}V_R + \left(\frac{(\mathrm{INT1}V_H + -\mathrm{INT1}V_R)\cdot(\mathrm{temp}_C + -\mathrm{temp}_R)}{(\mathrm{temp}_H + -\mathrm{temp}_R)}\right)$$

Back to [Equation 1], if we replace INT1V=1V by INT1V = $\mathrm{INT1V}_m$, we can deduce a finer temperature value as:

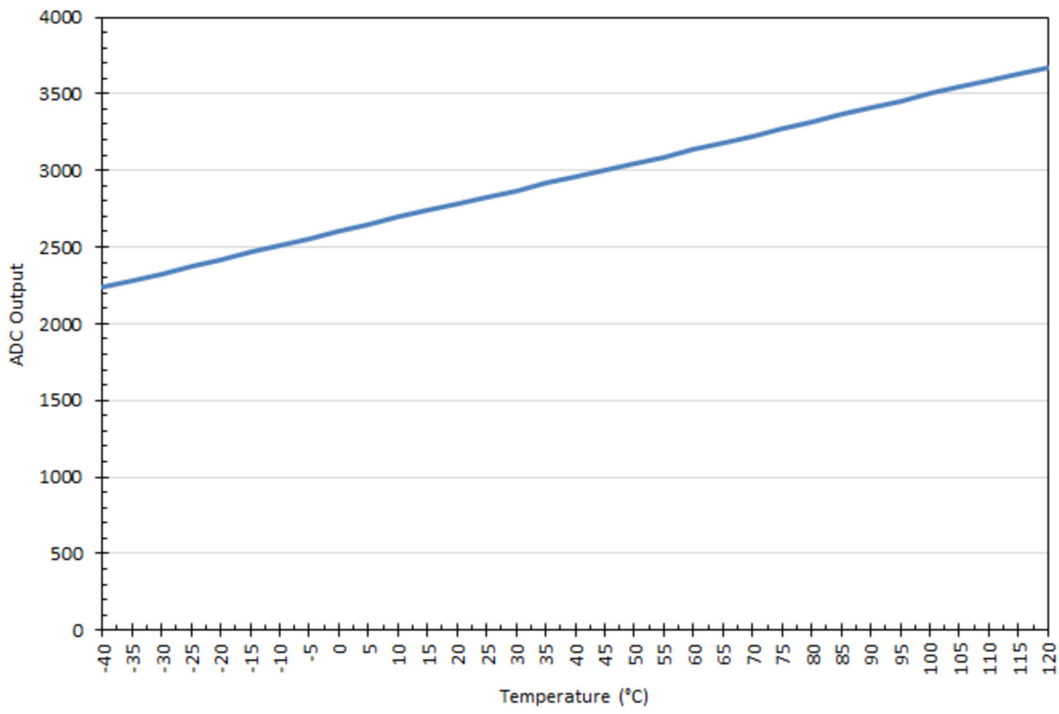
$$\mathrm{temp}_f = \mathrm{temp}_R + \left[\frac{\left\{\left(\mathrm{ADC}_m \cdot \frac{\mathrm{INT1}V_m}{(2^{12} + -1)}\right) + -\left(\mathrm{ADC}_R \cdot \frac{\mathrm{INT1}V_R}{(2^{12} + -1)}\right)\right\}\cdot(\mathrm{temp}_H \cdot \mathrm{temp}_R)}{\left\{\left(\mathrm{ADC}_H \cdot \frac{\mathrm{INT1}V_H}{(2^{12} + -1)}\right) + -\left(\mathrm{ADC}_R \cdot \frac{\mathrm{INT1}V_R}{(2^{12} + -1)}\right)\right\}}\right]$$

[Equation 1bis]

2.4. Temperature Sensor Output Across Temperature Range

The following plot shows the ADC output of temperature sensor, without any compensation, plotted over the temperature range. The measurements were recorded with the configurations specified in Temperature Log Row.

Figure 2-1 ADC Output vs. Temperature



From the preceding plot, it is observed that the output of temperature sensor is linear with increase in temperature.

Note:

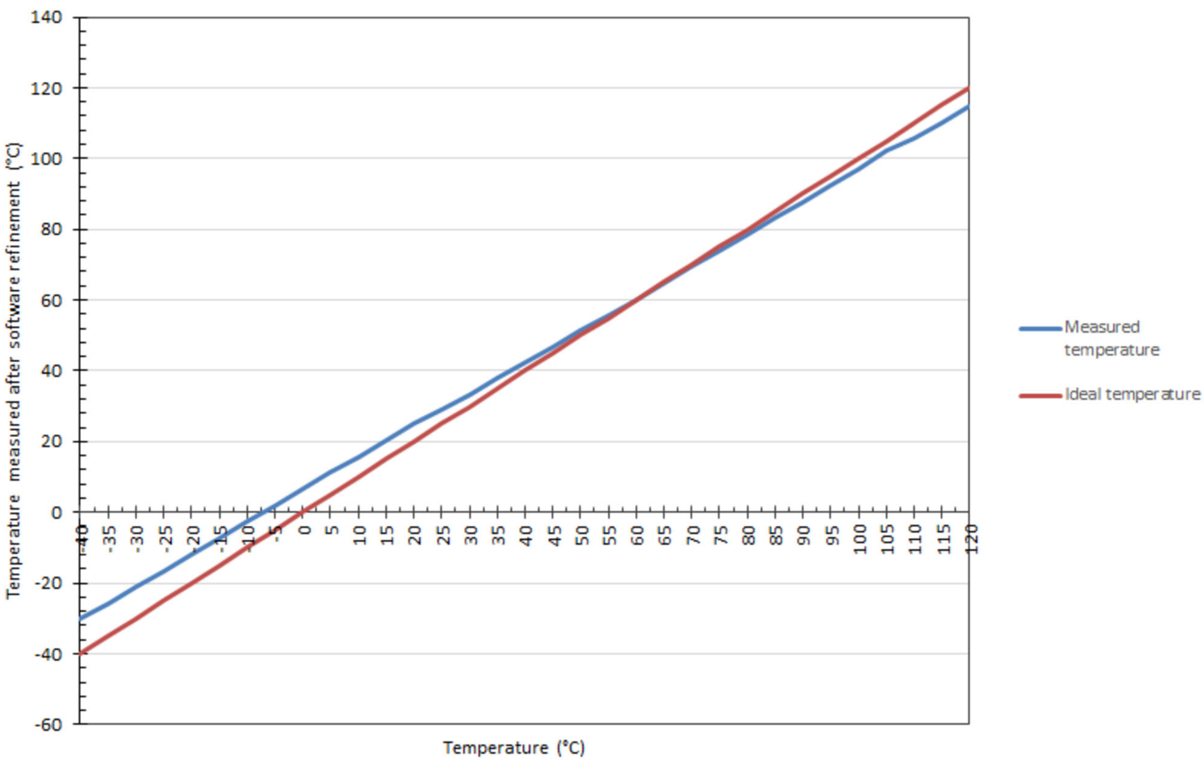
The graphical plots presented in the upcoming sections were all recorded using the same SAM R21 device to demonstrate the improvement of accuracy with each step of compensation. The improvement of accuracy was observed in 6 different SAM R21 devices.

2.5. Temperature Sensor Output Across Temperature Range After Internal Voltage Reference Compensation

The following plot shows the temperature value read from the sensor's output across the operating temperature range. This measurement uses the software based refinement method. These temperature measurement values are compensated for the variation in internal voltage reference.

For easier comparison, the ideal temperature has also been plotted.

Figure 2-2 Measured Temperature (after Internal Voltage Reference Compensation) vs. Temperature



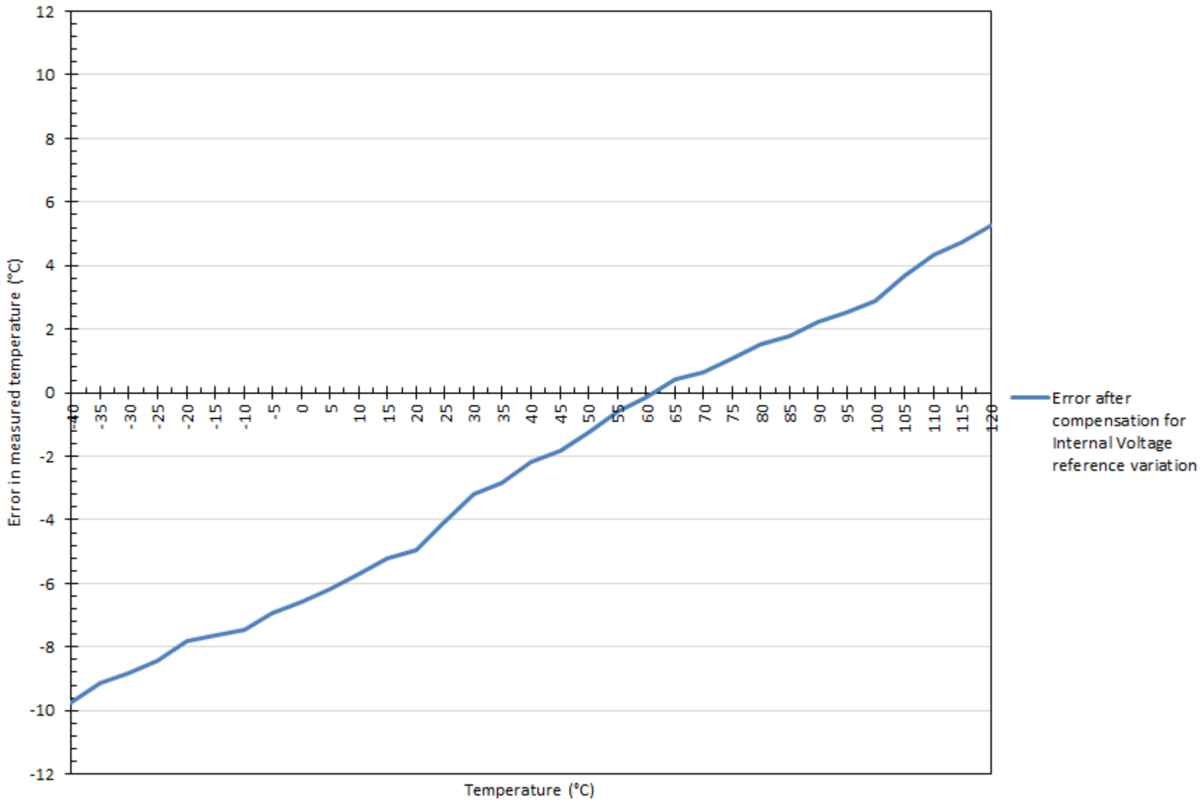
From this plot, we infer the variation of measurement error across the operating temperature range.

This error can be mathematically represented as,

$$T_{error} = T_{ideal} - T_{measured}$$

The error in measured temperature has been plotted in the following figure over the temperature range. It is observed from the plot, that the measurement error varies from -10°C to 5°C.

Figure 2-3 Error in Measured Temperature (after Internal Voltage Reference Compensation) vs. Temperature



3. Compensation for Offset and Gain Errors

The results from temperature measurements have offset and gain errors. These errors in the measured temperature value can be compensated by making calibration measurements at two known temperatures. From these measurements, offset and gain error in the temperature measurements can be calculated. Compensating the measured temperature for offset and gain error can result in very precise temperature measurements, sometime as accurate as ±3 °C. This section explains about the two-point calibration process in detail.

3.1. Two-point Calibration

To measure the temperature as accurate as possible, two calibration points are required. Let us consider these two points as T1 and T2. In the associated application, T1 and T2 are configured as 25°C and 80°C, respectively. The second calibration temperature must be chosen a little higher than the desired temperature range.

After compensating the internal voltage reference variation,

- Let ADC_T1 be the measured value of the temperature(in °C), at the first calibration point (T1)
- Let ADC_T2 be the measured value of the temperature(in °C), at the second calibration point (T2)

Note: The Two-point calibration can be applied to get an accurate temperature reading, only if the output from temperature sensor is linear.

3.1.1. Calculating the Offset Error

Let us plot a graph with X-axis as the temperature during the actual measurement and Y-axis as the temperature measured from ADC. Let the origin for the plot be (0,0). Plot a straight line defined by the two points (T1,ADC_T1) and (T2,ADC_T2). Let us mark a point (X,Y) in the line.

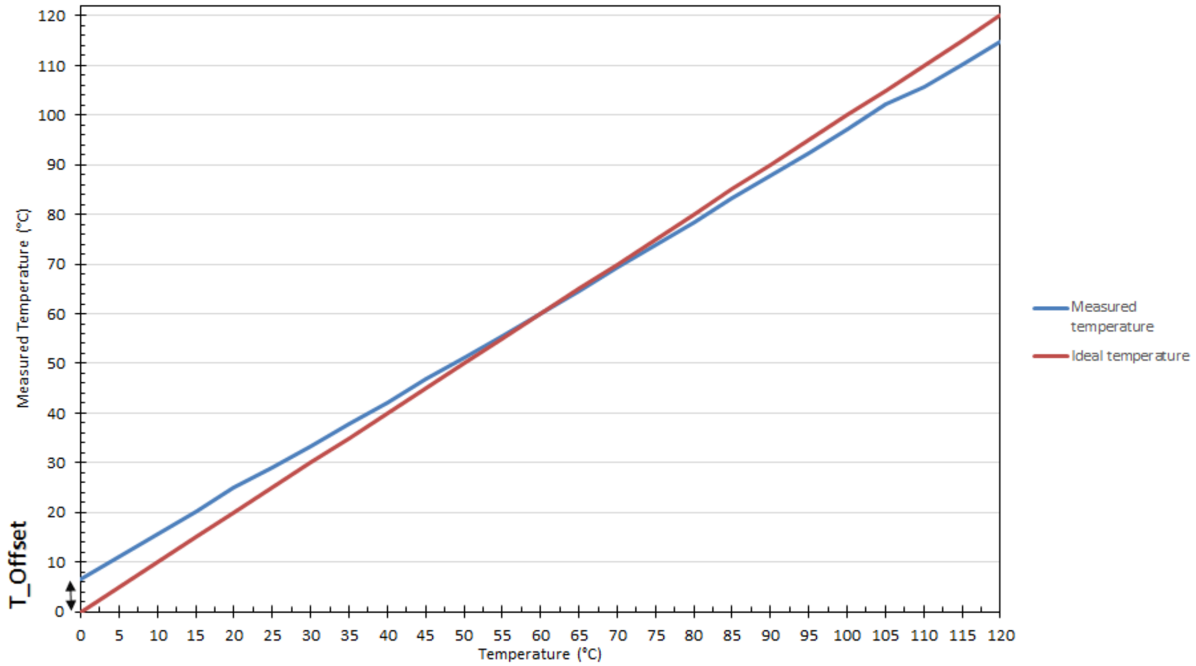
The slope of this line will be defined by the following equation,

$$Slope = \frac{(ADC\_T2 - ADC\_T1)}{(T2 - T1)} = \frac{(Y - ADC\_T1)}{(X - T1)}$$

From the above equation,

$$Y = ADC\_T1 + \frac{(X - T1)(ADC\_T2 - ADC\_T1)}{(T2 - T1)}$$

Figure 3-1 Measured Temperature (after Internal Voltage Reference Compensation) vs. Temperature



Ideally, at X=0, the value of Y must be equal to 0 (Y=0).

But, the above plot is affected by an offset error (T_Offset). This error is the distance between the origin of coordinates and the point where the line defined by the two calibration points intersects the Y-axis i.e.,Y-intercept of the line defined by the two calibration points, which can be found by substituting X=0 in the above equation.

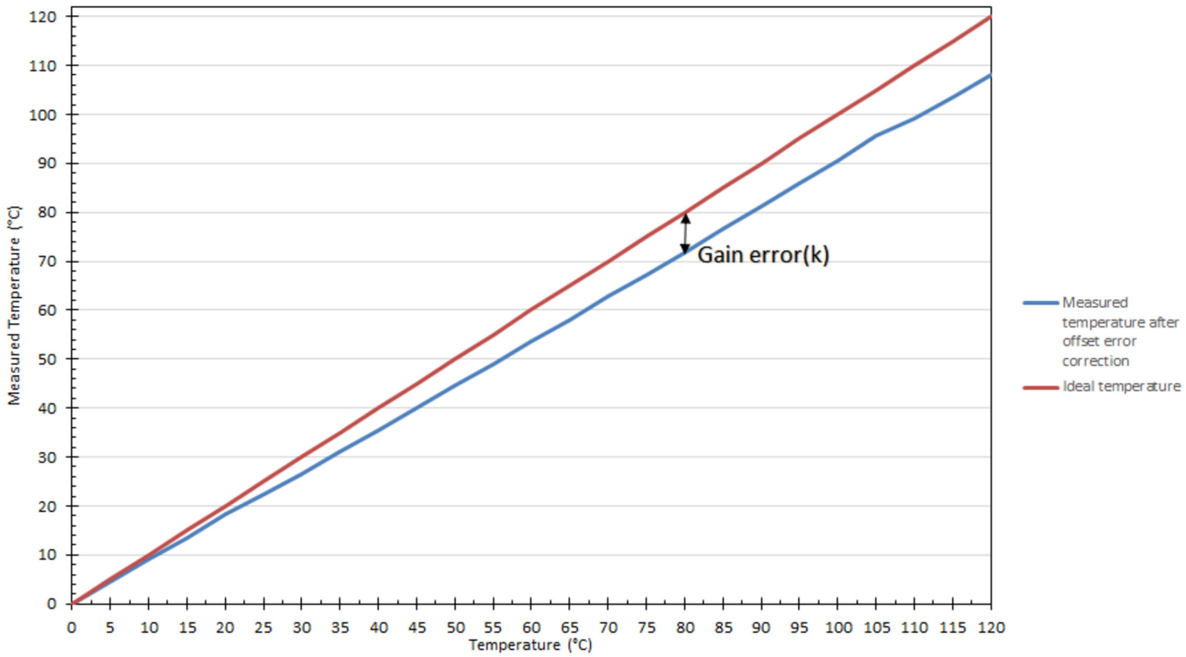
The resultant offset error can be calculated by,

$$T\_Offset = ADC\_T1 + \frac{(-T1)(ADC\_T2 - ADC\_T1)}{(T2 - T1)}$$

3.1.2. Calculating the Gain Error

The following figure shows the plot of measured temperature value after compensating the offset error. It can be observed that the plot is still affected by gain error.

Figure 3-2 Measured Temperature (after Internal Voltage Reference and Offset Error Compensation) vs. Temperature



It is necessary to find the gain factor(k) that affects the measured temperature value. Gain error is the ratio of Ideal Temperature value to that of the Measured temperature, after offset error compensation.

The gain error is mathematically represented by,

$$k = \frac{T2}{(ADC\_T2 - T\_Offset)}$$

Note: The value of k will not vary even if T2 is replaced with T1 and ADC_T2 is replaced with ADC_T1 in the above equation, as both the points lie in the same straight line.

3.1.3. Offset and Gain Error Compensation

The measured temperature value can be compensated for Offset and gain error using the following equation.

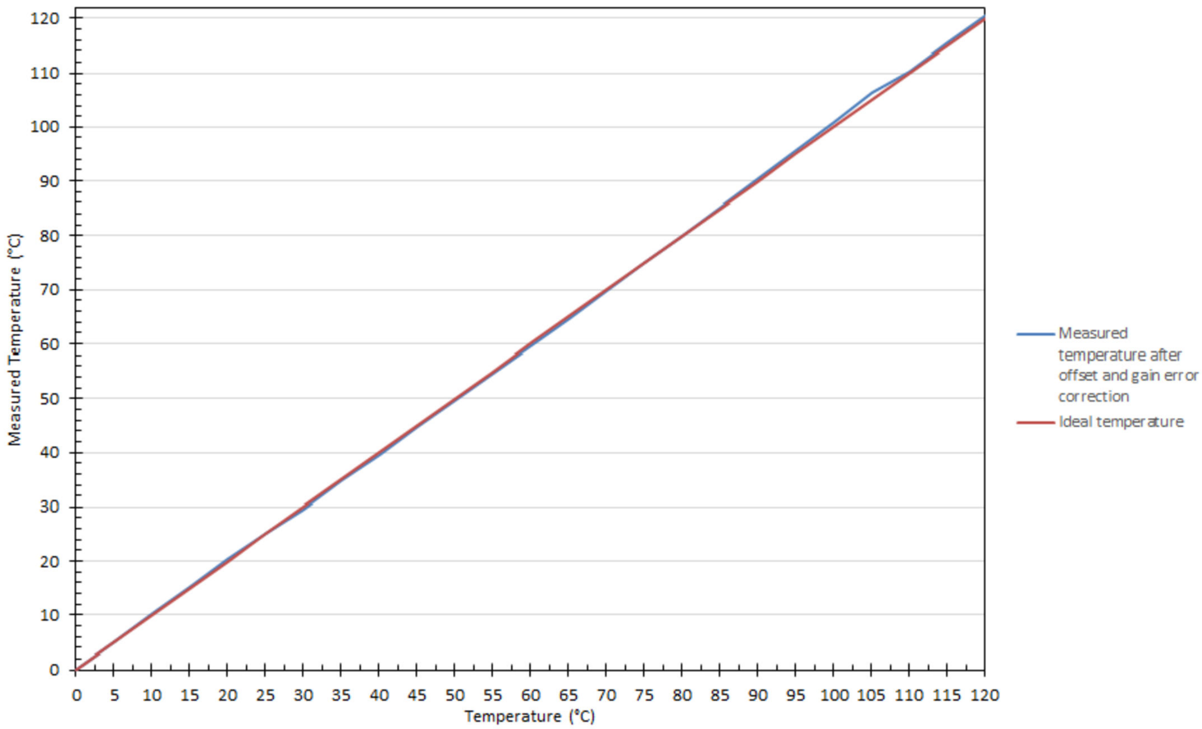
$$T\_c = (T\_ADC - T\_Offset) * k$$

Where,

- T_ADC is measured temperature value after internal voltage reference compensation
- T_c is the measured temperature value after internal voltage reference compensation, offset and gain error compensation

The following plot shows the measured temperature value after internal voltage reference compensation, offset and gain error compensation.

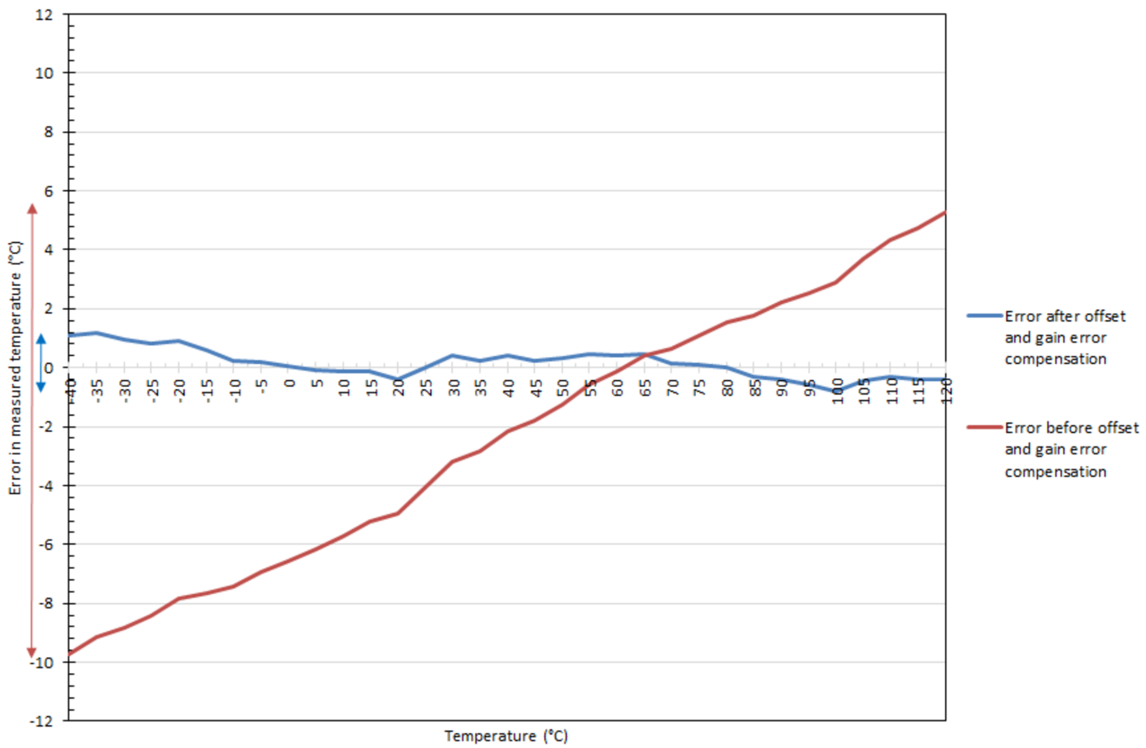
Figure 3-3 Measured Temperature (after Internal Voltage Reference compensation, Offset and Gain Error Compensation) vs. Temperature



3.1.4. Accuracy After Internal Voltage Reference, Offset and Gain Error Compensation

After compensating the measured temperature value for variation in internal voltage reference, offset error and gain error, the accuracy of the measured temperature value can be as high as ±3 °C. The error in measured temperature after offset and gain error compensation can be found from the below plot. The red arrow along the Y-axis indicates the span of the error before offset and gain error compensation. The blue arrow along the Y-axis indicates the span of the error after offset and gain error compensation.

Figure 3-4 Error in Measured Temperature (after Internal Voltage Reference, Offset and Gain Error Compensation) vs. Temperature



4. User Calibration

The firmware package available with this application note contains an Atmel Studio project (`AT14164.atsln`). This project shows the implementation of software-based refinement and two-point calibration process described in this application note. This project uses Atmel Software Framework (ASF V3.25.0). Steps to calibrate the internal temperature sensor of SAM R21 using this firmware is explained in the upcoming topics.

4.1. Hardware Setup

- 1 * SAM R21 Xplained Pro Board
- 1 * Temperature controlled chamber

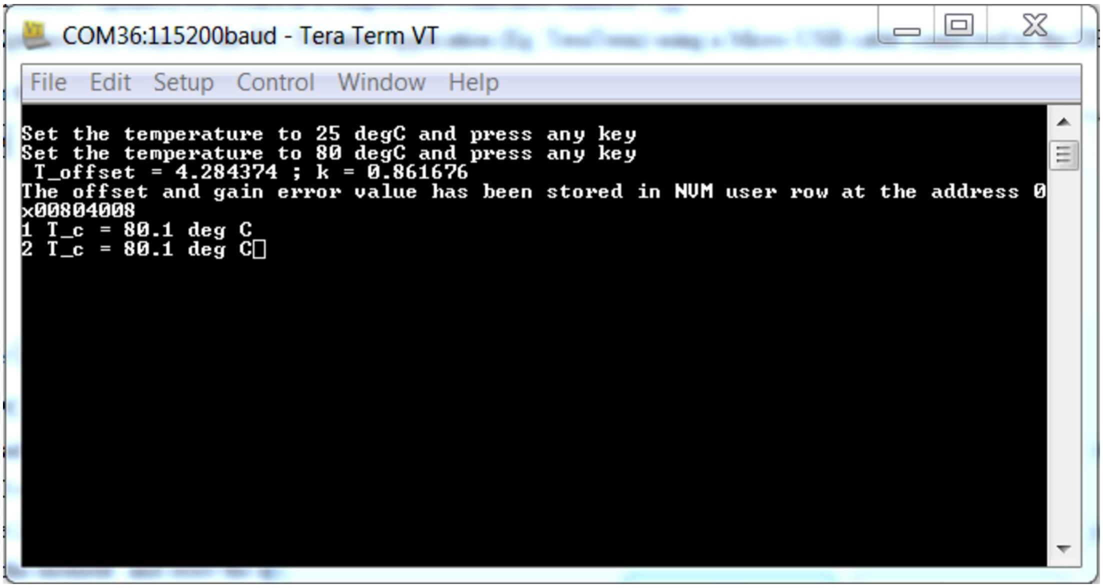
4.2. Launching the Application

To launch the default application,

1. Open the Atmel studio project (`AT14164.atsln`) in Atmel studio.
2. Compile the project and program the image generated from the project to the SAM R21 Xplained Pro board.
3. Place the programmed SAM R21 Xplained Pro board in a temperature controlled chamber.
4. Connect the SAM R21 Xplained Pro board to the PC-Terminal Application (e.g. TeraTerm) using a Micro-USB cable connected to the EDBG USB header of the board.
   Use the following setting for the serial connection with the PC-terminal:
   - BAUD RATE: 115200
   - PARITY: None
   - DATA BITS: 8
   - STOP BITS: 1
   - FLOW CONTROL: None
5. Press any key from the PC-terminal application, to start the application.
6. The application would load the data from the temperature log row of SAM R21.
7. Set the temperature to be 25°C and wait for some time (e.g. 10-20 minutes) for the board to settle.
8. Press any key from the PC-terminal application to proceed further. The application will read the temperature sensor's output at this moment and store the measured temperature to a variable `ADC_T1` in the application.
9. Set the temperature to 80°C and wait for some time (e.g. 10-20 minutes) for the board to settle.
10. Press any key from the PC-terminal application to proceed further. The application will read the temperature sensor's output at this moment and store the measured temperature to a variable `ADC_T2` in the application.
    The application will calculate the Offset (`T_offset`) and Gain(`k`) error in the measured temperature value. The calculated error values will be stored in NVM user row at the address 0x00804008.
11. The application will measure the temperature sensor output every 5 seconds and initiate the compensation process. The offset and gain error value will be read from the NVM user row. The read values will be used for compensating the measured temperature value for the offset and gain error. The measured temperature value after compensation will be printed in the terminal window.

The example PC-terminal window showing the message logs can be found below. In the terminal window, T_c corresponds to the measured temperature value after compensations (in °C).

Figure 4-1 Example PC-terminal Window



4.3. Application Overview

This topic describes the major functions used in the application project. The default application project runs with internal 8MHz oscillator of SAM R21.

The following code snippet shows the *main()* function of the application:

```
int main(void)
{
    system_init();

    delay_init();

    /*Serial Console configuration */
    configure_console();

    /*NVM configuration*/
    configure_nvm();

    /*ADC Configuration*/
    configure_adc();

    /*Enter any character from the terminal to start the
application*/
    getchar();

     /*Load the data from the Temperature Log row */
    load_temperature_log_row_data();

    /*Initiate the user calibration process*/
    adc_temp_sensor_calibration();

    while (1)
    {
        adc_temp_sensor();
```

```
        delay_s(5);
    }
}
```

calculate_temperature(): This function will compensate the measured temperature value for the variation in internal voltage reference. This function implements the procedure mentioned in the section Software-based Refinement of the Actual Temperature. This function is available in the *adc_temp.c* file of the firmware package.

load_temperature_log_row_data(): This function reads the contents of the temperature log row and stores it to global variables. This function is available in the *adc_temp.c* file of the firmware package. The stored values will be used in *calculate_temperature()* function.

nvm_set_calibration_fuse(): This function will write the NVM user row at address 0x00804008 with the offset and gain error calculated. This function is available in the *main.c* file of the firmware package. This function will mask the contents of NVM user row containing the fuse bits (0x00804000).

adc_temp_sensor_calibration: This function will calculate the values of offset and gain error on the measured temperature value by implementing the two-point calibration procedure mentioned in Compensation for Offset and Gain Errors on page 9. This function is available in the *main.c* file of the firmware package. The calculated values of offset and gain error will be stored in the NVM user row using the function *nvm_set_calibration_fuse()*.

adc_temp_sensor: This function will read the temperature sensor output at the given moment and call *calculate_temperature()* function to compensate the measured temperature value for the variation in internal voltage reference. This function is available in the *main.c* file of the firmware package. After this, the measured temperature value will be compensated for the offset and gain error, read from the NVM user row at address 0x00804008. The final result will be printed in the terminal window of the PC.

configure_adc(): The function configures the ADC module with the configuration parameters mentioned in the Temperature Log Row except for the averaging configuration. In this function, ADC is configured to operate in averaging mode with 64 samples to decrease the volatility in the temperature sensor output. This function is available in *main.c* file of the firmware package associated with this application note.

The application can be modified to change the first and second calibration point. In the default application project, first calibration point is fixed to 25°C and second calibration point is fixed to 80°C. This can be modified by changing the values of the variables T1 and T2 in *adc_temp_sensor_calibration()* function of *main.c* file.

If the operating frequency of the application has to be modified, ADC prescaler configuration in *configure_adc()* function needs to be modified such that the configuration of ADC mentioned in Compensating Internal Voltage Reference Variation on page 4 is not modified. In addition to this, other parameters like flash wait states also needs to be taken care by the user.

5. Revision history

Doc Rev.	Date	Comments
42561A	11/2015	Initial document release

Enabling Unlimited Possibilities®

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**