Freescale Semiconductor
Application Note

Document Number: AN4904
Rev. 0, 04/2014

Kinetis KM3x MCU PT100 Sensing

by: Freescale Semiconductor, Inc.

1 PT100 sensing

This document demonstrates how to precisely measure the temperature difference between two PT100 sensors using the Kinetis MKM34Z128 processor. The measurement is to be used for a heating meter design. The heating meter design requires very precise temperature sensing in the range of 0°C to 200°C of absolute temperature, and the temperature difference from dt> 3°C, where the relative differential temperature error is better than 3%. A temperature difference error for low temperature difference (~3°C) may be measured with an error below 1.5%. This MCU offers four very precise 24-bit sigma delta AD converters which may be used for temperature sensing. For accuracy, all components/parameters causing drifts should be avoided.The PT100 sensor resistivity is rated against a known normal resistor RN. A current mirror is used to ensure the same current is flowing through the sensor and the normal resistor. This topology doesn't require precise current or voltage source, only a stable normal resistor is needed. To calculate the temperature from the resistivity provided by the PT100 sensor, a 3rd order regression polynomial is used. All calculations are done in 32-bit signed fractional fixed-point arithmetic.

Contents

2 Measurement principles

The current mirror formed around R1, R2 and Q1, ensures that the rate of the current flowing through the normal resistor R_N , and of that flowing through the PT100 sensor, is roughly given by the rate of the R1 and R2 resistors. Provided that the U_{PT} and U_{RN} voltages are measured by the SD ADC, the actual resistivity of the PT100 sensor may be calculated by equation [9].

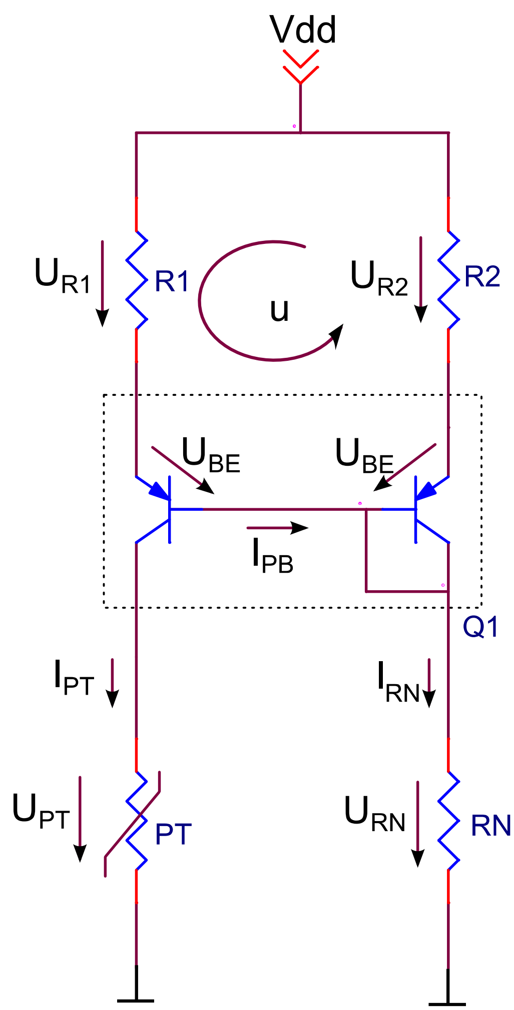

Figure 1. Heating and measurement

The thermocouple double transistors are used in the circuit, so we can assume that the “u” voltage loop equation

Eqn. 1

$$U_{R1} + U_{BE} - U_{BE} - U_{R2} = 0$$

may be rewritten as [2], as U_{BE} has the same value for both transistors.

Eqn. 2

$$(1_{R1} = 1_{R2}) \bullet R^2 / R^1$$

Currents through the PT100 sensor and normal resistor R_N are defined by equations [3] and [4]

Eqn. 3

$$1_{R1} = 1_{PT} + 1_B$$

Eqn. 4

$$1_{R2} = 1_{RN} - 1_B$$

If equation [3] is derived, substituting I_{R1} with [2]

Eqn. 5

$$1_{R2} \times R_2 / R_1 = 1_{PT} + 1_B$$

Eqn. 6

$$1_{R2} = R_1 / R_2 (1_{PT} - 1_B)$$

Then the right-hand side of equations [4] and [6] are used, and the current replaced by voltage / resistivity

Eqn. 7

$$U_{RN} / R_N - 1_B = R_1 / R_2 - U_{PT} / R_{PT} - R_1 / R_2 \bullet 1_B$$

Eqn. 8

$$R_{PT} = \frac{U_{PT}}{U_{RN}/R_1 + (1_B(R_1 - R_2))/R_2}$$

And, if I_B is neglected for simplicity, and substitution of $k = (R_1 * R_N) / R_2$, equation [8] is simplified to

Eqn. 9

$$R_{PT} = k \times U_{PT}/U_{RN}$$

From equation [9], it is clear that the actual temperature sensor PT100 resistivity may be calculated if the voltage on the PT100 and voltage on the normal resistor are known. Of course, the values of resistors R_1 R_2 *and* R_N have to be known.

3 How to measure a temperature difference by the PT100 pair

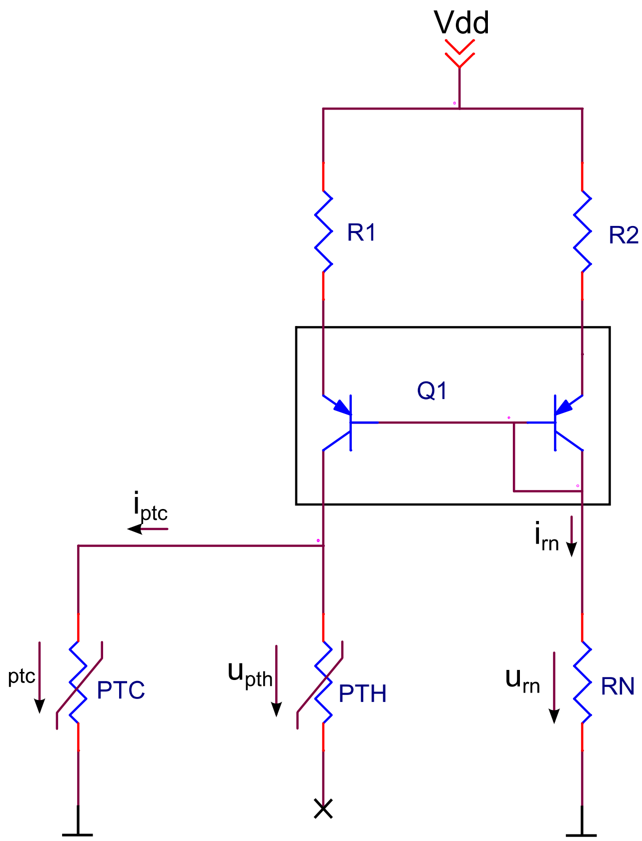

Figure 2. Measurement step one - PTH measurement

The measurement of a single temperature by the PT100 is demonstrated in Section 2, "Measurement principles. Measurement of a temperature difference by the two PT100 sensors is shown in the following figure. There are two temperature sensors present. The PTC and the PTH are for cold and hot water respectively. The PTH sensor is grounded, while the PTC sensor is floating (Hi-Z). Supply voltage Vdd is applied. Then, after voltage stabilization, the voltage values of PTC and URN are measured.

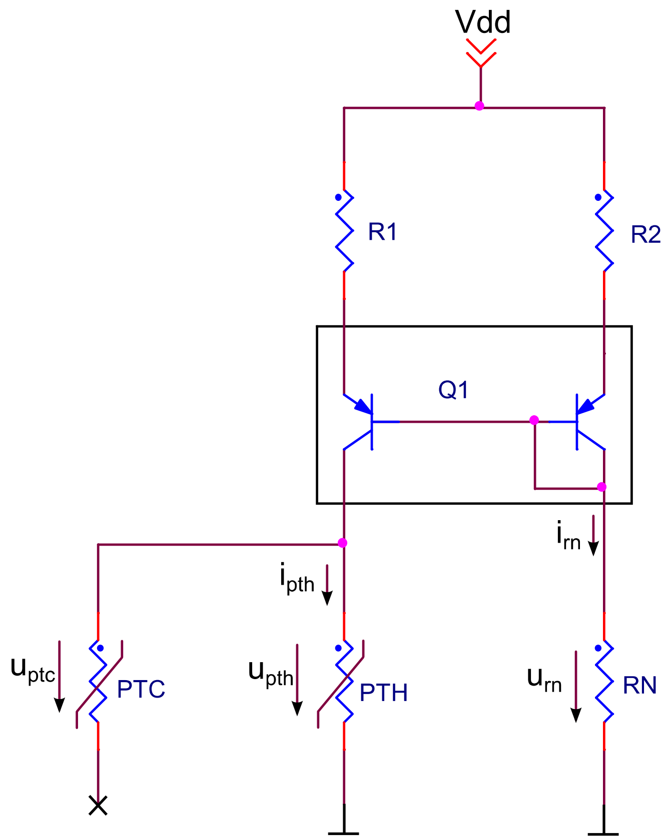

Figure 3. Measurement step two - PTC measurement

The second step is similar. As demonstrated in Figure 3, sensor PTH is left floating, PTC is grounded, and after stabilization, the voltages UPTC and URN are measured.

Always take care to first ground one of the sensors, PTC or PTH, before the supply voltage Vdd is applied, to avoid an out of range voltage on the ADC inputs.

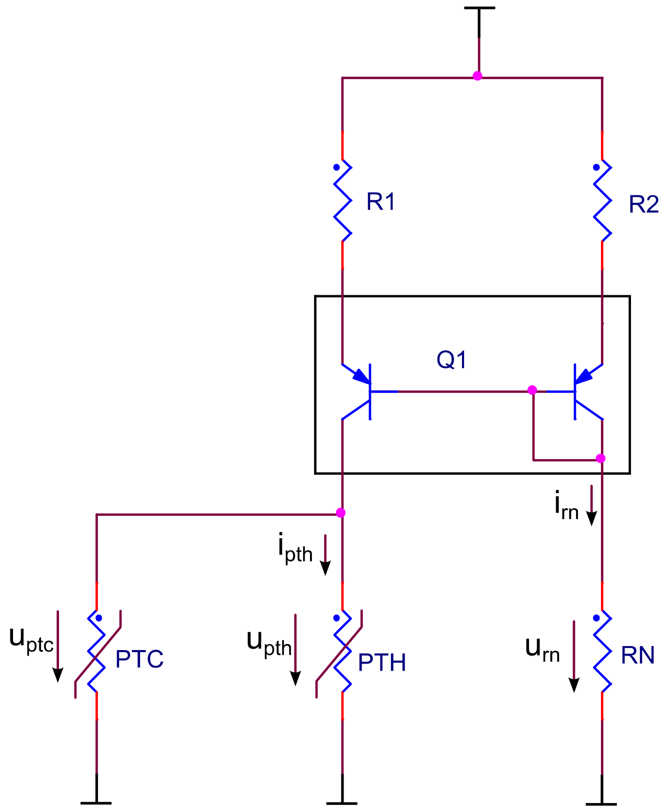

Figure 4. Measurement step three - Shut off measurement

The final step is to put the circuit into the off state, as shown in Figure 4. To avoid a voltage floating on the ADC inputs and pins during a non-measurement period, all pins in the circuit are grounded.

The ADC offset measured during the shut off state may be used for offset compensation later on, by subtracting the measured offset value from the conversion while the circuit is supplied. This simple action lowers the ADC offset error.

4 The real circuit and its parameters

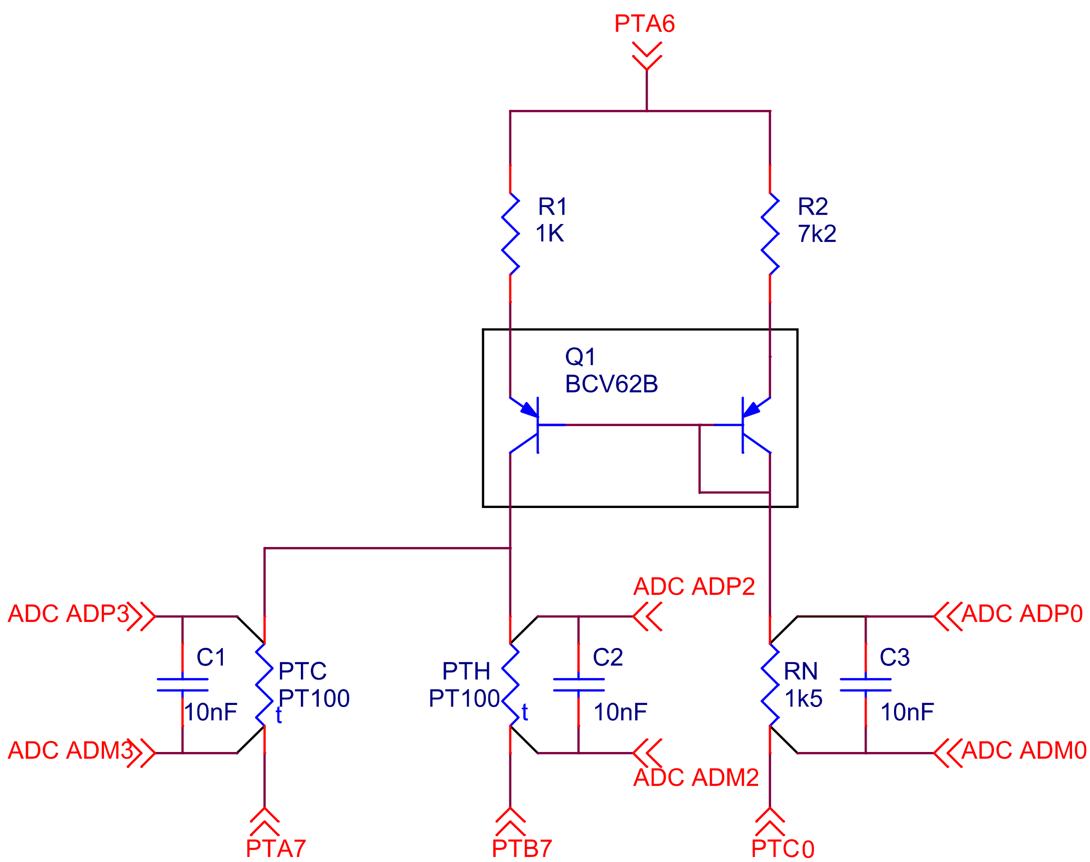

Figure 5. Measurement circuit

The power supply is sourced from the PTA6 pin. Temperature sensors PTC and PTH are grounded by PTA7 and PTB7 respectively. The normal resistor is grounded by PTC0.

The thermal sensors are sensed by the differential sigma delta AD converter; channel 3 for the PTC and channel 2 for the PTH sensor. The normal resistor is sensed by ADC channel 0.

The resistors R2 and RN set the current through R_N to roughly 260uA. U_{RN} is roughly 400mV, so that the full ADC range (Vdd <= 2.3V as there is a voltage drop on the pins) UEB = 0.6V.

Current flowing through the PT100 sensor is set, by [2], to 2mA.

The PT100 temperature range of 0 to 200°C means 100 Ohm to 175 Ohm, so that the voltage on the sensor is in the range 0.2V to 0.35V.

The reason the measured voltage UPT must always be lower than U_N is hidden in the equation [10]. In the following calculation, fractional arithmetic is used and the divisor must always be lower than the dividend (UPT < URN).

NOTE

It would be better to use pins from the same MCU port, otherwise the pins may have different voltages when set to output low, due to a different grounding potential. Connect ADC ch2 and ch3 (those without PGA) to the thermal sensors to maintain accuracy.

5 What accuracy can be achieved

There are several sources of measurement errors. An initial error could be introduced by a bad operational point due to resistor value variations, initial ADC calibration errors, and PT100 sensor errors as well. Any initial error may be easily compensated for by calibration.

Another source of error is thermal drift. There are thermal drifts of resistors and transistors in the current mirror, and also offset drifts of the ADC converter. Errors caused by a thermal drift have to be carefully handled. The following paragraph discusses errors caused by the thermal drift of the resistors used in the current mirror.

5.1 Thermal drift error

Looking at equation [9], constant k = (R1*RN) / R2 is a combination of the resistors used in the current mirror. The thermal coefficient of the resistors will directly influence coefficient k and, therefore, the calculated measured sensor resistivity. A careful selection of resistors with a low thermal coefficient is needed.

The R1, R2 drift may be partially compensated for if the thermal drift coefficient has the same polarity. The RN drift directly influences the measurement error. Simply, the lower the thermal drift coefficient, the lower the measurement error.

On the other hand, if we measure the thermal difference by the two PT100 thermal sensors rather than absolute temperature, the relative error of the thermal difference is compensated for, as coefficient k is applied in both equations.

$$R_{PTH} = k \times U_{PT} / U_{RN}$$

$$R_{PTC} = k \times U_{PTC} / U_{RN}$$

In summary, the absolute value of temperature calculated from the measured thermal sensor resistivity is influenced by the thermal drift of the resistors used.

The error in the thermal difference calculated from both sensors is partially compensated for due to the fact that both temperatures suffer by the same error caused by the k coefficient.

The ADC offset thermal drift also directly affects resistivity measurement. There is a simple method of compensating for the ADC offset thermal drift.

The ADC offset thermal drift may be directly measured by running the ADC conversion while no voltage is applied to the thermal sensors. The value measured is then subtracted from the ADC measurement during a normal operation. These simple steps help to remove the offset drift error introduced by the ADC.

5.2 Initial operational point error

Any initial operational point error is corrected during the calibration process. The measured value of resistivity suffers from offset and gain errors from the sources mentioned above. Resistivity of the PT100 sensor is calculated by equation [9] and then calibration is done by adding gain G'_{PT} and offset O_{PT} to the equation.

$$R_{PT} = G'_{PT}\left(k \times \frac{U_{PT}}{U_{RN}}\right) + O_{PT}$$

k constant and gain G'_{PT} may be merged to $G_{PT} = G'_{PT} * k$

$$R_{PT} = G_{PT} \times \frac{U_{PT}}{U_{RN}} + O_{PT}$$

Eqn. 10

5.3 Accuracy measured – temperature difference

Accuracy is measured in such a way that precise resistors are used instead of the PT100 sensors to simulate a given temperature. Resistivity is then calculated from the measured voltages.

The graph in the picture 6 shows the measurement error of the resistivity difference, dt = th-tc, when identical resistors are applied on both th and tc, and the expected difference is 0 Ohm.

On the x-axis, the value of the resistor applied on both the hot and cold sensors.

On the y-axis, the measured resistivity difference [Ohms]. The expected value is 0.

Measurement is done for a resistivity interval of 100-200 Ohms. The MCU and measurement circuit are cooled / heated from -10°C to 50°C, while the precise resistors are at a stable temperature.

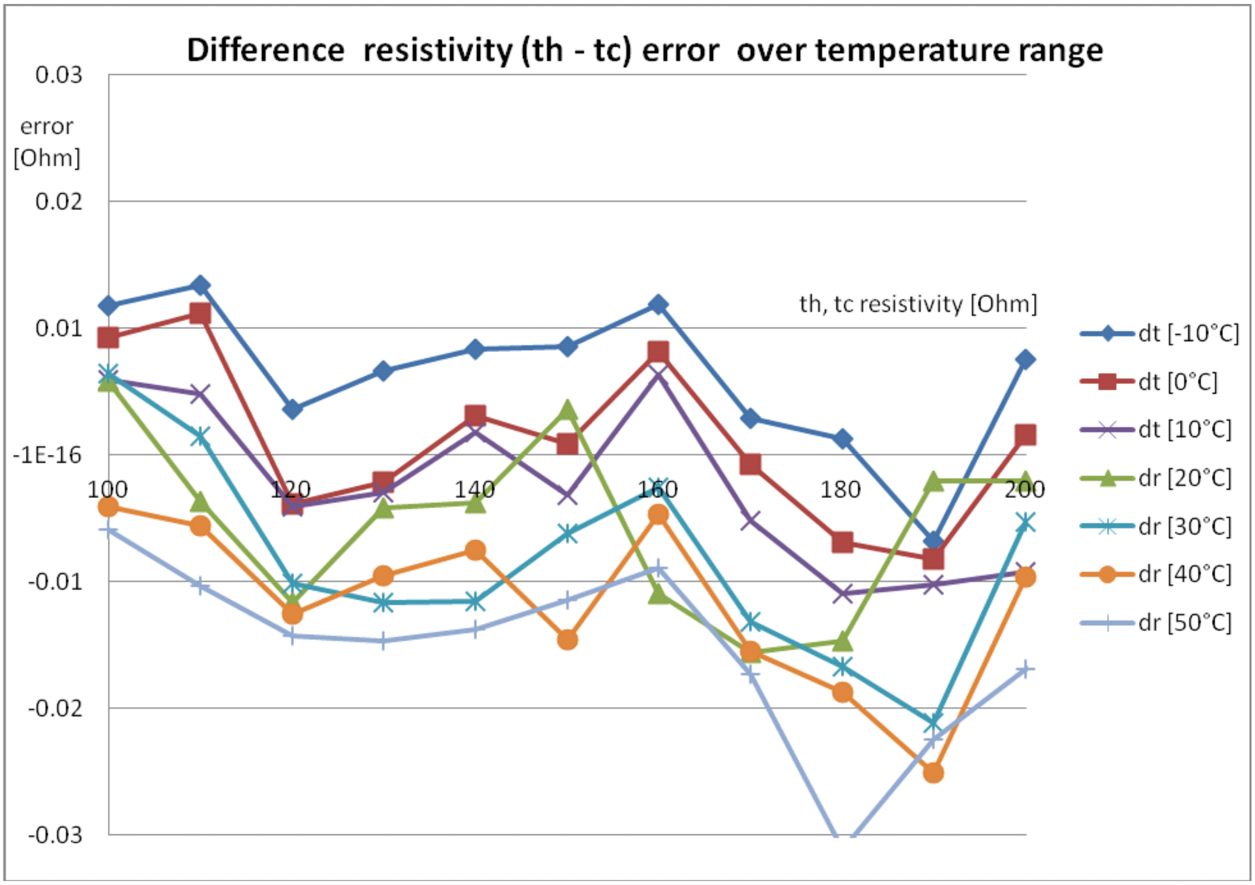

Figure 6. Resistivity difference on the hot and cold sensors, measurement errors = th-tc [Ohm]

5.4 Accuracy measured – absolute resistivity error

Here, absolute resistivity is measured. Precise resistors are used instead of the PT100 sensors to simulate a given temperature. Resistivity is then calculated from the measured voltages.

On the y-axis, there is the error of the resistivity measured

error = Rmeasured -Rreal

On the x-axis, there is resistivity applied.

Clearly evident is the influence of the thermal coefficient of resistor R_N and the drift of the ADC converter.

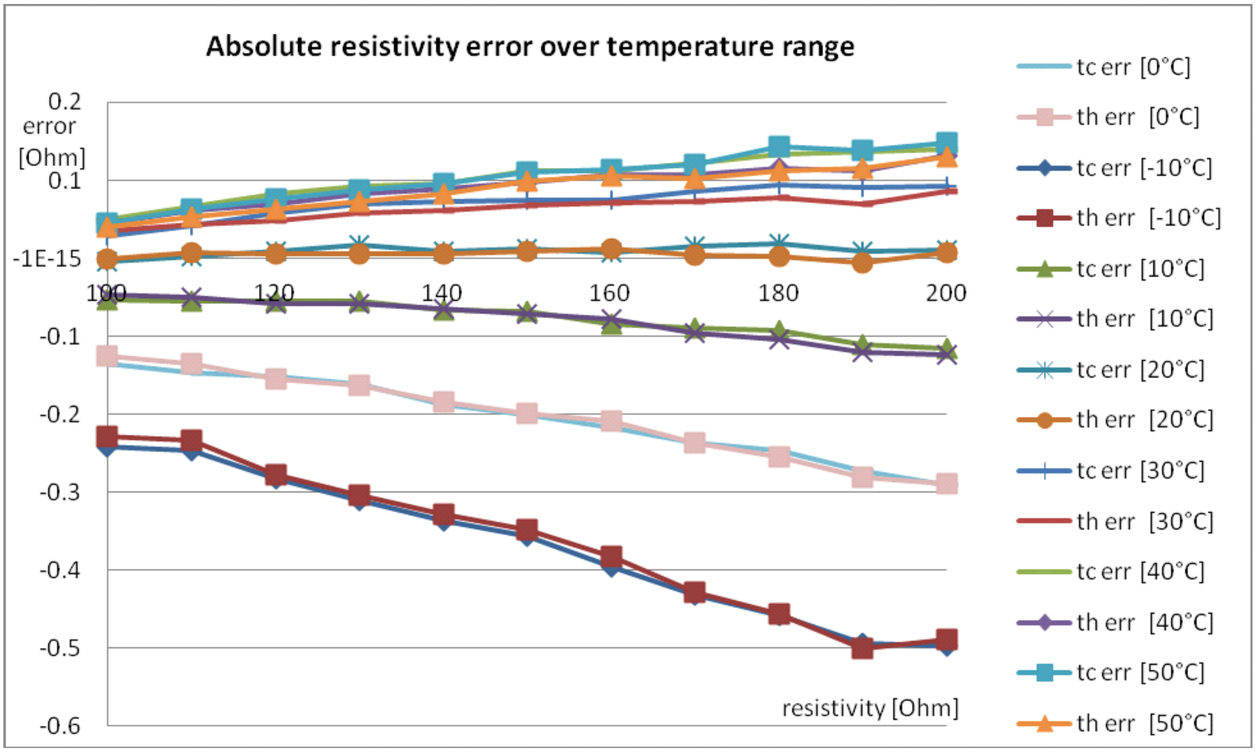

Figure 7. Absolute resistivity error on the hot and cold sensors

6 How to get temperature when resistivity is known

6.1 Normalization

Values in the result registers of the AD converters represent the voltage on the PT100 temperature sensor and voltage on the normal resistor R_N. Those values are not scaled with their physical representation – degrees of centigrade. What's more, function T=f(R) is not linear and it needs to be calculated by regression, or as the solution of a quadratic equation.

For this kind of calculation, fractional fixed-point mathematics are used to our advantage, and because fractional arithmetic uses values <-1; 0.9999> range, it is necessary to define the rate of measured to physical units in an appropriate way.

Normalization rates are defined as follows, because 32-bit signed fractional arithmetic is used, and 2^31 represents the full scale:

Resistivity: 220 Ohm ~ 0x7fffffff (2^31)

Temperature: 600°C ~ 0x7fffffff (2^31)

The rates chosen ensure that normalized values don't cross the valid interval <-1; 0.9999> if measurement is in the defined range of 0°C to 200°C.

6.2 Calibration

Initial errors must be compensated for by calibration, as described in Section 5.2, "Initial operational point error.

Precise resistors of known value, for example 100 Ohm and 200 Ohm, are applied to simulate PT100 sensors. Voltages UPT and URN are measured by the ADC for both resistor values.

Values of the gain GPT and the offset OPT in the equation [10] are calculated from the measurement of two known values of resistors applied instead of the PT100 sensor.

To match the stated normalization values, the values of resistivity measured after calibration must equal their theoretical counterparts, as shown in the following table.

Table 1. Calibration points

Applied resistivity [Ohm]	R_{PT} theoretical value [integer]	Fractional representation
220 (full scale)	R_{PT220}: 0x7fffffff	0.999999
200	R_{PT200}: 0x745D1744	0.909090
100	R_{PT100}: 0x3C3C3C3B	0.470588

To figure out the values of gain and offset, it is necessary to solve two equations with two unknowns derived from equation [10].

$$R_{PT} = G_{PT}\left(\frac{U_{PT}}{U_{RN}}\right) + O_{PT}$$

U_{PT}: voltage on the sensor measured by the ADC [output code]

U_{RN}: voltage on the normal resistor measured by the ADC [output code]

R_{PT}: expected normalized value of resistivity calculated after calibration

G_{PT}: applied gain value

O_{PT}: applied offset value

The equation is valid and measured at 2 working points – where 100 and 200 ohms are applied – thus, UPT100 and URN100 are values measured when a 100 Ohm resistor is used instead of a sensor, while UPT200 and URN200 are measured when a 200Ohm resistor is applied. Those values are used to write 2 equations with two unknowns.

$$R_{PT100} = G_{PT} \times X_{100} + 0_{PT}; \quad X_{100} = \frac{U_{PT100}}{U_{RN100}}$$

$$R_{PT200} = G_{PT} \times X_{200} + 0_{PT}; \quad X_{200} = \frac{U_{PT200}}{U_{RN200}}$$

$$O_{PT} = \frac{R_{PT200} \times X_{100} - R_{PT100} \times X_{200}}{X_{100}}$$

The PT100 temperature is not a linear function of resistivity. The PT100 sensor resistivity out of temperature function is defined by the equation

$$r = 100(1 + at + bt^2); \quad a = 0.0039083; \quad b = -0.0000005775$$

where *t[°C]* is temperature and *r[Ohm]* resistivity.
This equation is valid only for temperatures above zero degrees of centigrade.

Here we are looking for the inverse equation. We need to figure out temperature out of known resistivity. The most precise way to enumerate temperature is to use a quadratic formula. The drawback of this method is the need to use a square root calculation. Another way to enumerate temperature is to use a regression function. The input parameter to the regression function is the normalized calibrated temperature, and the function output is the normalized temperature.

To achieve good regression accuracy, a 3-rd order regression function is used. The regression function coefficients *a*, *b*, *c*, *d* may be found in Matlab.

$$t(r) = a \times r^3 + b \times r^2 + c \times r + d$$

a = 0.016547476356113

b = 0.054142784657457

c = 0.878861127710399

d = -0.412227297382249

Figure 8 shows the absolute error of the regression function. Regression is calculated by 32-bit fractional mathematics.

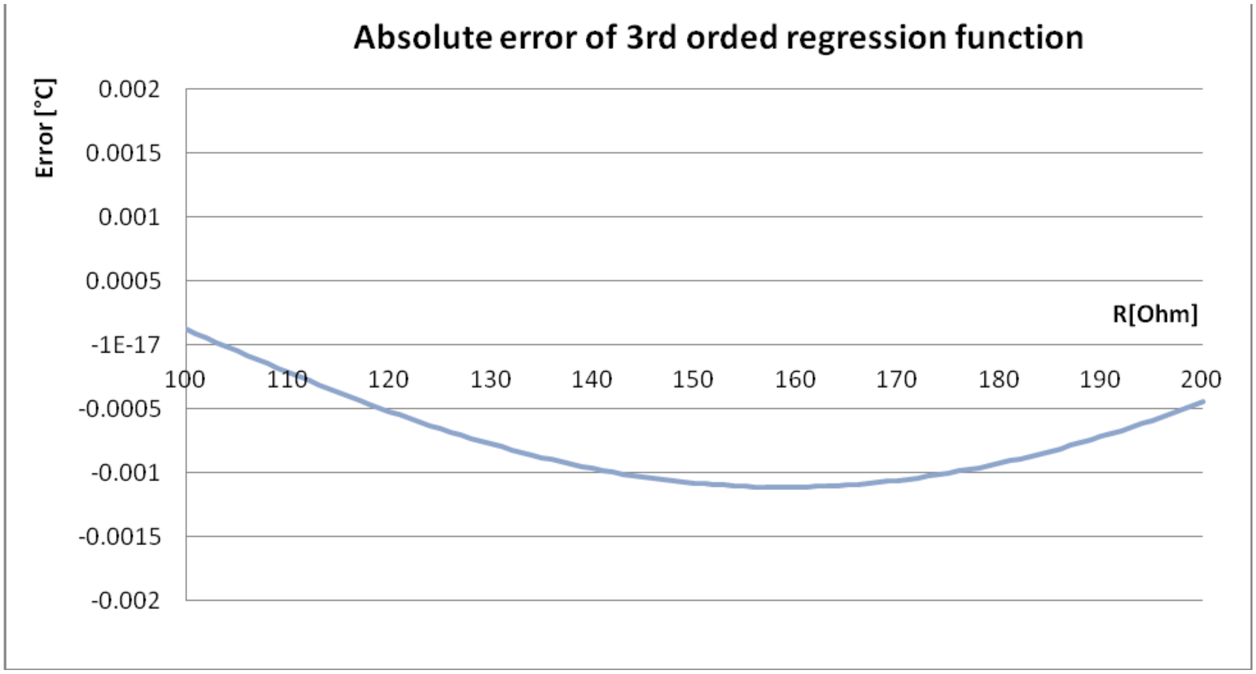

Figure 8. Absolute error of the regression function

The calculated temperature is normalized to 600°C, due to the fact that we need all the regression coefficients (a,b,c,d) to be lower than 1 for calculation in fractional mathematics.

Temperature: 600°C ~ 0x7fffffff (2^31) and the matching regression function:

$$t_{norm}(r) = 0.01654748 \times r^3 + 0.054142785 \times r^2 + 0.878861128 \times r - 0.412227297$$

Finally, to show the temperature of the sensor is found only by multiplication of the normalized temperature with the normalization rate, $t[°C] = 600 * t_{norm}$

7 Conclusion

The MKM34Z128 processor, with four 24-bit high precision sigma delta converters, has a set of features ideal for designing a heating meter application. This application note demonstrates the accuracy of temperature measurement. It is evident that the MKM34Z128 24-bit sigma delta converters have a high accuracy that enable a precise differential temperature measurement.

The maximal relative accuracy of the temperature difference, Dt = 3°C measured, is 1.5%. This allows the construction of a heating meter by OIML R75.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number: AN4904
Rev. 0
04/2014