Application Report

SBOA173A – January 2017 – Revised December 2019

Optimizing Remote Temperature Sensor Design

David Vaseliou
Emmy Denton

ABSTRACT

Many high-performance systems rely on accurate temperature measurements to optimize performance and ensure reliable operation. The temperature readings can be used to adjust temperature-sensitive elements, such as A/D converters and high-end displays, or the readings may be used to monitor system health and prevent overheating. While modern processors have built-in temperature measurement, these are generally not as accurate as an external temperature sensor and overall system performance may suffer if an external sensor is not used. As system density increases, the use of external sensors that measure many temperature points also help simplify the system design for thermal management design.

This application report discusses the design considerations for successfully sensing the temperature of a highly integrated system using a digital remote temperature sensor. The application report specifically focuses on the TMP468, but the information can be applied to other digital remote temperature sensors such as the TMP431, TMP451, and TMP461. Remote temperature sensors sense the junction temperature of either a NPN or PNP bipolar junction transistor (BJT). The BJT can be an integrated transistor in a MCU, GPU, ASIC, FPGA, or a discrete transistor such as the common 2N3904 NPN transistor or 2N3906 PNP transistor. The use of remote temperature sensors is common in telecom equipment (switches and routers), servers, automotive infotainment, Advanced Driver-Assistance Systems (ADAS), and high-end displays. System design can be quite challenging in highly integrated systems because noise and BJT process variations can cause errors. This application report gives an overview of remote temperature sensors, discusses error sources, and shows how to mitigate system impact of sensor errors. Layout considerations are also discussed.

Contents

1 System Description 3
2 Troubleshooting a Noisy System 15
3 Conclusion 16

List of Figures

1 TMP468 System Diagram 3
2 Possible Remote Junction Configurations 4
3 V_{BE} Measurement 5
4 Δ_{VBE} Measurement With Two Currents 5
5 Example Data set of TMP468 Temperature Error Under Default Settings 8
6 Example Data Set of TMP468 Temperature Error Using Offset Register Calibration 9
7 Example Data Set of TMP468 Temperature Error Using n-factor Correction Register Calibration 9
8 Example Data Set of TMP468 Temperature Error Using Calibration Tool Settings 9
9 Transistor-Connected PNP Grounded Collector 10
10 Accuracy Without Beta Variation Compensation 10
11 Accuracy With Beta Variation Compensation 10
12 Δ_{VBE} Measurement With Three Currents Cancels Errors Caused by Series Resistance 11
13 TMP468 Input Stage Simplified Schematic 12
14 Leakage Resistance Effect on TMP468 Accuracy 12
15 Schematic for Noise Tests 13
16 Noise Aggressor Waveform With a 100-µs Time Base 13
17 Noise Aggressor Waveform With a 1-ms Time Base 13
18 TMP468EVM Baseline Noise Level Measurement Without Noise Injection 14
19 TMP468EVM Noise Level Measurement With Noise Injection 14
20 Routing of D+ and D- as Single-Ended Traces on Multiple Layers 15
21 Routing as Differential Pairs on Multiple Layers 15
22 Isolating Noise Coupling Into Remote Transistor Traces 15
23 Isolating Noise Coupling Through Power Supply or Other Sources 15

List of Tables

1 Example Data Collection With Default Settings 8

Trademarks

All trademarks are the property of their respective owners.

1 System Description

Figure 1 shows a circuit diagram of a remote temperature sensing system. The circuit includes the TMP468 device that monitors nine temperature zones using the internal (local) temperature sensor, and eight remote transistors with the associated filter circuitry (R_{S1}, R_{S2}, and C_{DIFF}). Shown in the diagram is a two-wire I²C or SMBus compatible system management controller that reports the zone temperatures from the TMP468 to a console, or coordinates system cooling and protection. Figure 1 shows overtemperature shutdown circuitry that takes direct hardware action to shut down the system based on the $\overline{\text{THERM}}$ outputs of the TMP468. The TMP468 includes two limits for each temperature zone that are associated with the $\overline{\text{THERM}}$ and $\overline{\text{THERM2}}$ outputs. These limits digitally compare to the current temperature readings and activate the $\overline{\text{THERM}}$ and $\overline{\text{THERM2}}$ outputs.

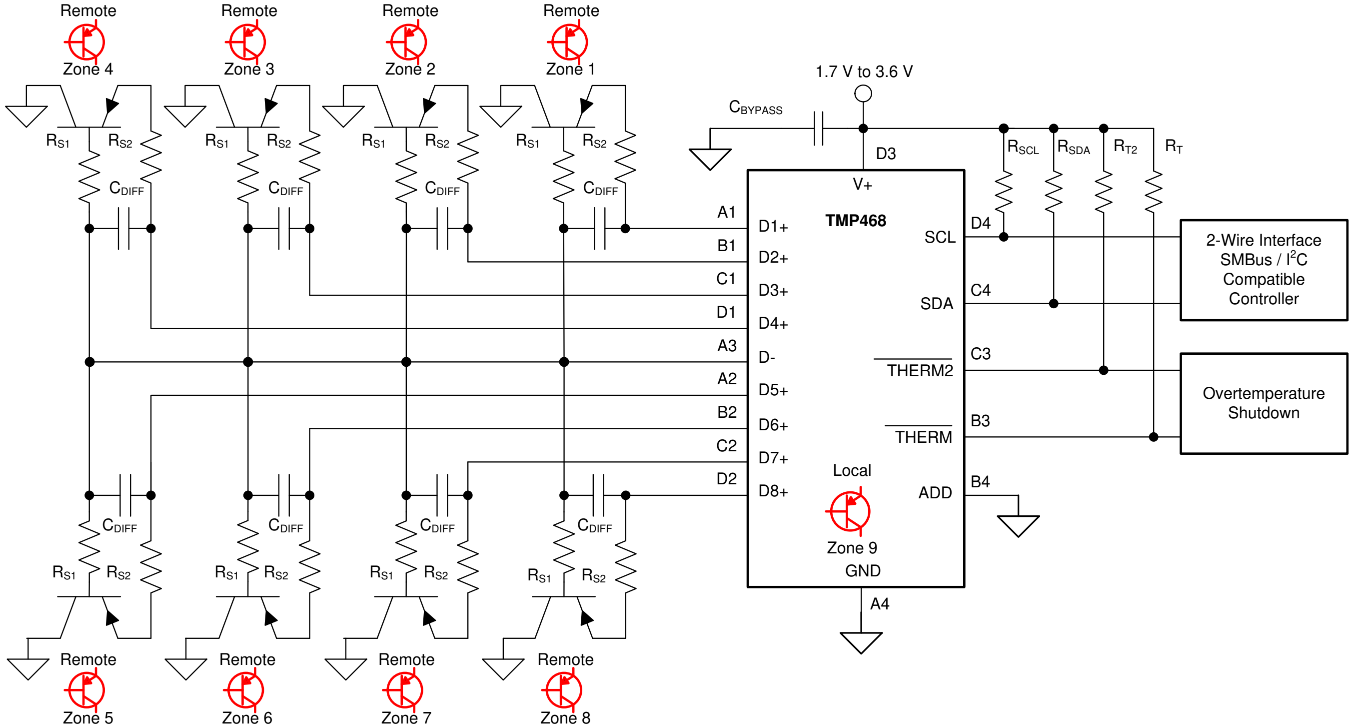


Figure 1. TMP468 System Diagram

1.1 Remote Temperature Sensing Overview

The base emitter junction of a BJT has a very predictable transfer function that is dependent on temperature. Remote temperature sensors use this principle to measure the temperature of an external transistor. There are four possible BJT configurations, as shown in Figure 2. The selected configuration is typically dependent on availability and the type of task. In most cases, remote temperature sensors measure the junction temperature of another device, such as a high-power processor, FPGA, or ASIC. The most common bipolar transistor found in CMOS processes is a substrate PNP with a collector that is tied to ground or substrate. To measure the case temperature of a device or board temperature, use a discrete transistor, as shown in Figure 2. Complicated systems require robust thermal management solutions with several temperature zones for protection and prevention of thermal runaway. Thus, the TMP468 measures the junction temperature of multiple highly integrated devices, and the temperature of discrete transistor junctions placed throughout the system to generate a thorough temperature profile.

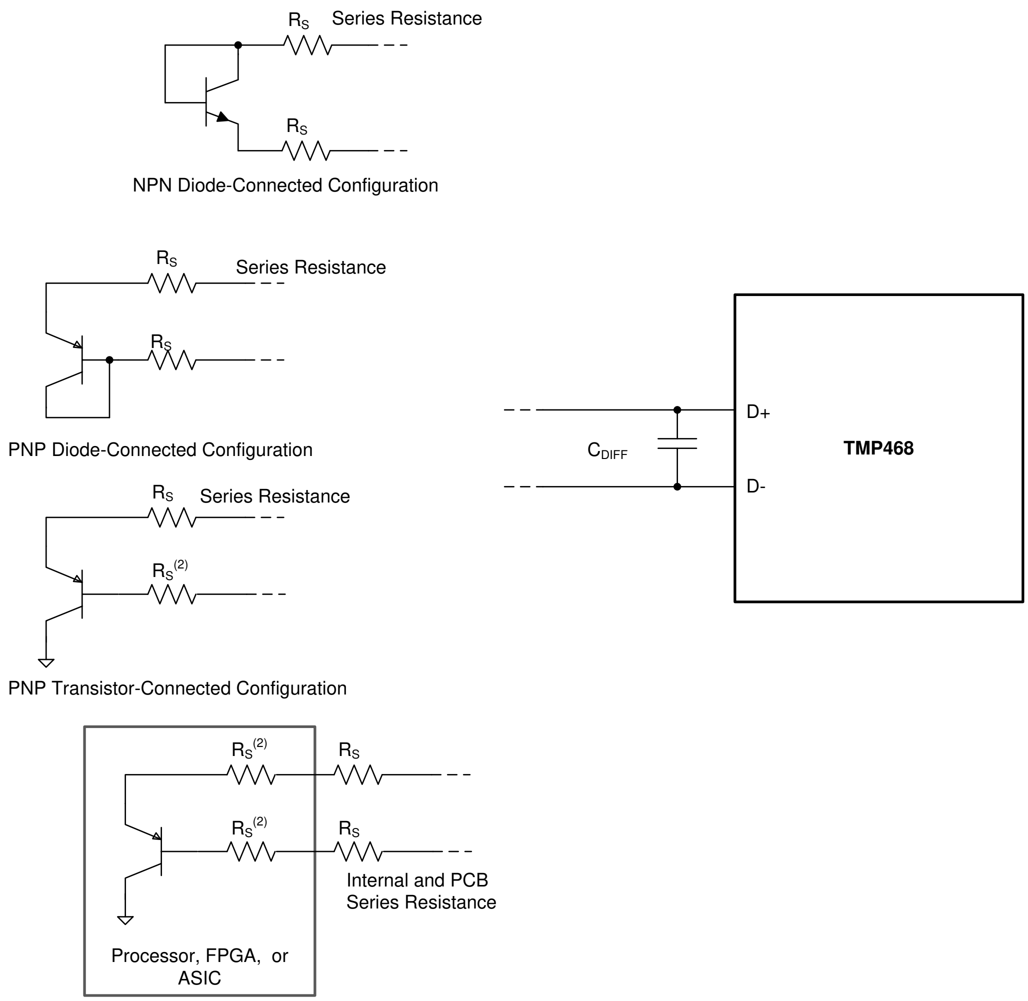


Figure 2. Possible Remote Junction Configurations

Figure 3 shows the most straightforward method of measuring a transistor base emitter voltage by forward biasing the transistor junction with a fixed current.

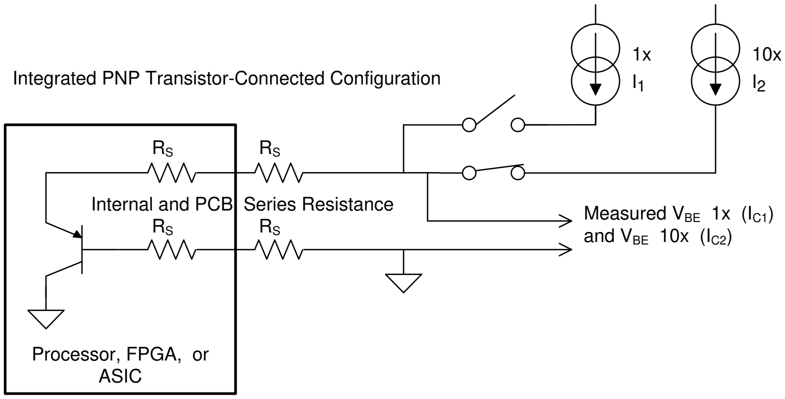


Figure 3. V_{BE} Measurement

The standard Ebers-Moll model equation describes the function of this circuit. Equation 1 shows the simplified equation for the collector current.

$$I_C = I_S e^{\frac{qV_{BE}}{\eta kT}}$$

where
- I_C is the transistor collector current
- I_S is the transistor reverse saturation current
- q is the charge of an electron (1.60217662 × 10^{-19} coulombs
- k is Boltzmann's constant (1.3806485 × 10^{-23} m^2 kg s^{-2} K^{-1})
- η is the ideality factor of the transistor (n-factor)
- T is the temperature of the transistor in degrees Kelvin
- V_{BE} is the base emitter voltage drop (1)

Solving for V_{BE} results in Equation 2:

$$V_{BE} = \frac{\eta kT}{q} \ln\left(\frac{I_C}{I_S}\right) \quad (2)$$

N-factor and the reverse saturation current terms have process dependencies, and can vary widely from one transistor type to another. The device manufacturer can control the collector current, but precision can become costly. This method is not widely used, since error variability in the range of ±9°C has been observed. For these reasons, the two current method approach (as shown in Figure 4) has gained popularity.

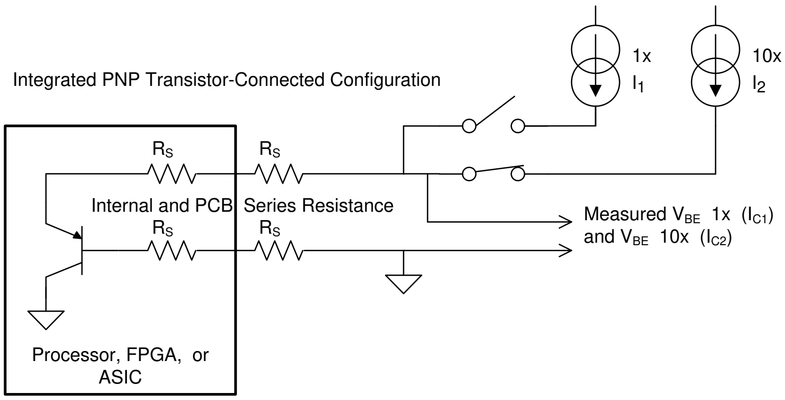


Figure 4. Δ_{VBE} Measurement With Two Currents

The two current method takes advantage of cost-effective circuits that provide very precise current ratios, which cancel out the effects of the process variation of reverse saturation current. Using the two current method, the difference of two voltage measurements is used to determine the temperature, as shown in Equation 3:

$$\Delta V_{BE} = V_{BE1} - V_{BE2} \quad (3)$$

Substituting Equation 2 into Equation 3 results in Equation 4:

$$\Delta V_{BE} = \frac{\eta kT}{q} \ln\left(\frac{I_{C1}}{I_S}\right) - \frac{\eta kT}{q} \ln\left(\frac{I_{C2}}{I_S}\right) \quad (4)$$

Simplifying Equation 4 results in Equation 5:

$$\Delta V_{BE} = \frac{\eta kT}{q} \ln(r)$$

where

$$r = \frac{I_{C1}}{I_{C2}} \quad (5)$$

Common values for the ratio (r) ranges from 10 × to 32 ×, yielding ΔV_{BE} voltages in the range of tens of millivolts. Such small signal levels can be sensitive to noise pickup and series resistance. In addition, the process dependency of the n-factor, if not accommodated, can introduce additional errors. Notice that the Ebers-Moll model solution depends on the ratio of the collector current, yet most remote temperature sensors only control the emitter current. Hence, variation in BJT β with different emitter current levels can cause errors if excessive variation occurs.

1.2 Remote Temperature Sensing Error Sources

This section covers the major sources of errors involved with remote temperature sensing, such as n-factor variation, BJT β variation, series resistance, and noise injection. It also describes how to overcome them.

1.2.1 Variation of N-Factor

Compensating for n-factor differences is simple if the diode manufacturer specifies the n-factor in the respective data sheet. If the n-factor of the transistor is not specified, the manufacturer may be able to provide the n-factor value by a special request. The TMP468 includes an independent register for each remote input that can be set to a value between 0.950205 and 1.073829. A simple equation provided in the TMP468 data sheet (Equation 6) can be used to determine the actual register value that must be programmed. The result of the equation is in decimal, and the n-factor correction register format is in two's complement with a range of –128 to +127. If the result of Equation 6 is not a integer, the value must be rounded to the nearest whole number to input into the n-factor correction register.

$$N_{ADJUST} = \left(\frac{1.008 \times 2088}{\eta_{EFF}} \right) - 2088 \quad (6)$$

where

- N_{ADJUST} is the decimal value required by the n-factor adjust register
- η_{EFF} is the n-factor of the BJT target

To calculate the n-factor of the target BJT from the TMP468 register decimal value, use Equation 7:

$$\eta_{EFF} = \left(\frac{1.008 \times 2088}{2008 + N_{ADJUST}} \right) \quad (7)$$

Some remote temperature sensors do not include n-factor adjust registers. These devices are typically calibrated for a 2N3904 or an MMBT3904 transistor. An offset register is typically provided in these devices to allow for a one point offset calibration that compensates for errors.

Temperature errors associated with n-factors of different processors or transistor types may be reduced in a specific temperature range of concern through use of software calibration. Typical n-factor specification differences cause a gain variation of the transfer function, so the center of the temperature range must be the target temperature for calibration purposes. Equation 8 can be used to calculate the required temperature correction factor (T_{CF}) to compensate for a target n-factor that differs from the 2N3904 transistor.

$$T_{CF} = \left(\frac{\eta_{SENSOR} - \eta_{PROCESSOR}}{\eta_{SENSOR}} \right) \times (T_{CR} + 273K) \quad (8)$$

where

- η_{SENSOR} is the n-factor that the remote temperature sensor is calibrated for (usually 1.008 or 1.003 for the 2N3904 transistor)
- $\eta_{PROCESSOR}$ is the new n-factor value of the processor or transistor
- T_{CR} is the temperature value at the center of the temperature range
- T_{CF} is the temperature to compensate for a target n-factor

The correction factor must be directly added to the temperature reading that the remote temperature sensor produces. For example, when using a remote temperature sensor that is calibrated with an n-factor of 1.008 to measure another BJT with a typical n-factor of 1.003, for a temperature range of 60°C to 100°C, the correction factor would calculate to:

$$T_{CF} = \left(\frac{1.008 - 1.003}{1.008} \right) \times (80 + 273.15) = 1.7517 \quad (9)$$

Therefore, 1.7517°C must be added from the remote temperature sensor temperature readings to compensate for the differing typical n-factor target.

1.2.2 Compensating for Errors Using Lab Measurements

If the n-factor, beta, and other sources of error are unknown, a simple bench evaluation can determine the viable calibration settings. First, test the target transistor with the default n-factor and offset settings of the remote temperature sensor over the desired temperature range. It is recommended to test multiple transistors and multiple remote temperature sensors to accommodate for part-to-part variation. After collecting the data, create a table using the expected temperature and the device temperature readings. Table 1 shows the format of an example data set and Figure 5 shows an example data set using default settings.

Table 1. Example Data Collection With Default Settings

Device Number	Expected Temperature	Device Reading
1	70	75.4
1	80	85.6
1	90	96
1	100	106.3
2	70	75.6
2	80	85.9
2	90	96.2
2	100	106.4
3	70	75.4
3	80	86
3	90	96.3
3	100	106.7

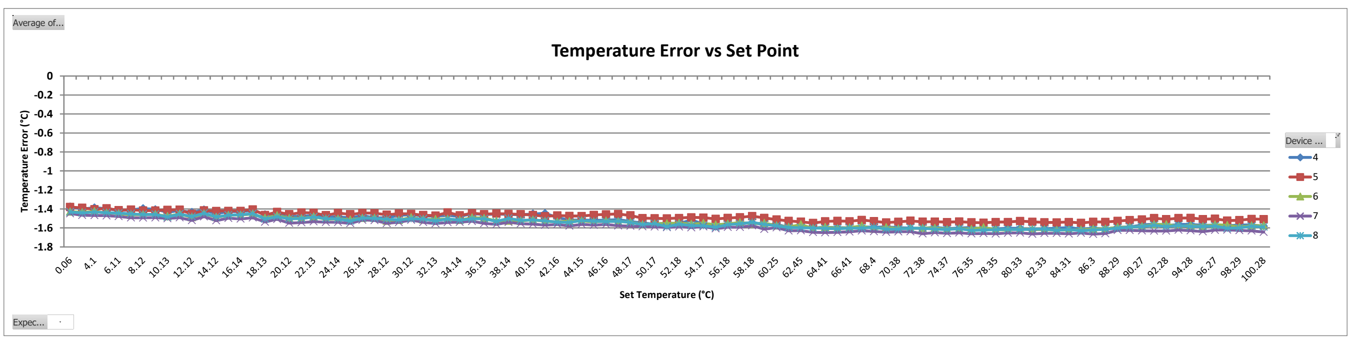


Figure 5. Example Data set of TMP468 Temperature Error Under Default Settings

After collecting the temperature data, calibration can be done using the offset register, n-factor correction register, or a combination of the two.

Using just the offset register can be a straightforward calibration method, but will leave any slope error that may be present in the data set. This is done by finding the temperature error at the point of interest and adding or subtracting the error at that point. As Figure 6 shows, the point at the center of the data set has been zeroed out, but there is still a slope error over the temperature range.

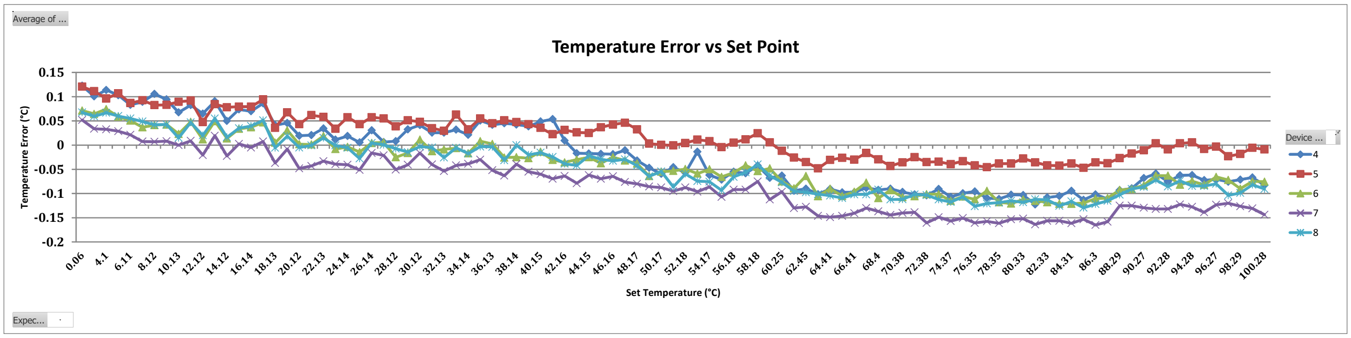


Figure 6. Example Data Set of TMP468 Temperature Error Using Offset Register Calibration

Another method is using just the n-factor correction register. Using the n-factor register can accommodate errors due to differing n-factors, but will not account for other error types that are present. Figure 7 shows the calibration results using just the n-factor correction register.

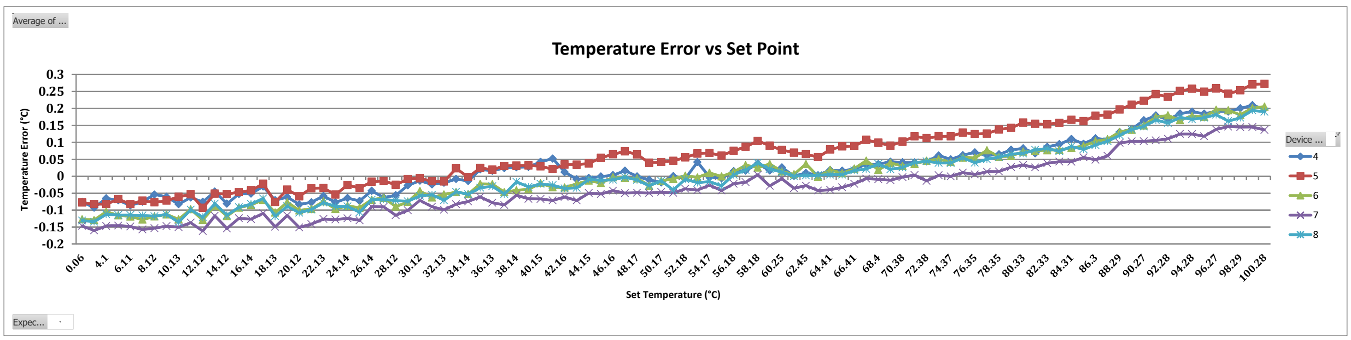


Figure 7. Example Data Set of TMP468 Temperature Error Using n-factor Correction Register Calibration

Lastly, both the n-factor and offset register can be used to correct for system errors. To do this, the data can be input into the *Remote Temperature Sensor Calibration Tool* on ti.com. Download and unzip the folder and follow the instructions in the excel tool. The tool runs an optimization algorithm that calibrates for both slope and offset errors in the system and outputs the optimal offset and n-factor correction settings for the data set. Figure 8 shows the calibration results using the optimal combination of n-factor and offset settings.

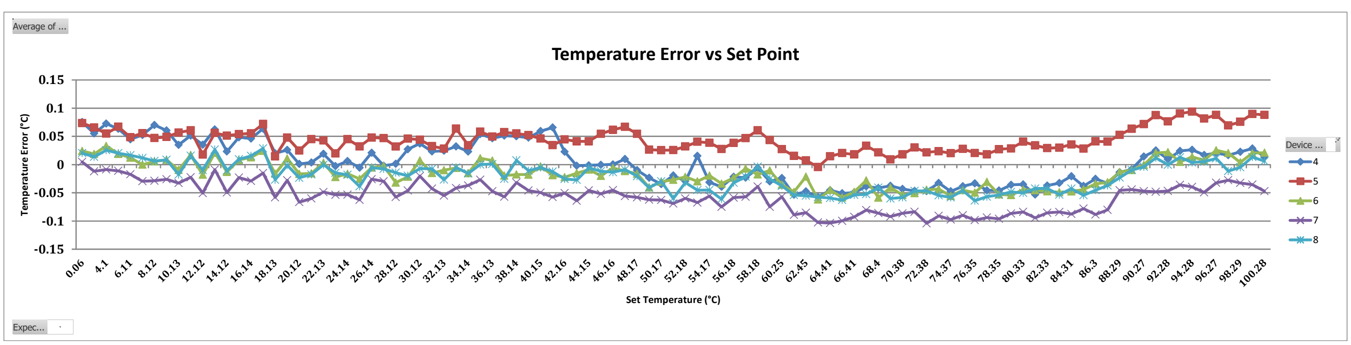


Figure 8. Example Data Set of TMP468 Temperature Error Using Calibration Tool Settings

1.2.3 Variation of BJT β

Maintaining an accurate current ratio is very critical since the current ratio directly effects the temperature reading. Errors in the current ratio appear as a temperature error. Equation 5 is dependent on the collector current. In the case of an integrated PNP diode where the collector is tied to ground (as shown Figure 9), the remote sensor forces the emitter current of the transistor.

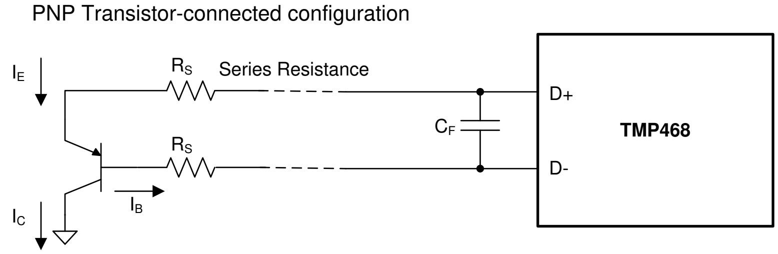


Figure 9. Transistor-Connected PNP Grounded Collector

There is a direct relationship between the emitter current and the collector current as shown in Equation 10. If β varies with the level of the forced emitter current, the collector current ratio is effected. Since the reciprocal of beta plus 1 effects the ratio (see Equation 11) large betas cause a negligible error. When beta approaches one, the change in beta with different currents has a greater effect on the ratio.

$$I_E = I_C + \frac{I_C}{\beta} = I_C \times \left(1 + \frac{1}{\beta}\right) \quad (10)$$

$$\frac{I_{E1}}{I_{E2}} = \frac{I_{C1} \times \left(1 + \frac{1}{\beta_1}\right)}{I_{C2} \times \left(1 + \frac{1}{\beta_2}\right)} \quad (11)$$

As processor geometries decreased, the beta of the substrate PNP decreased. Around the 90-nm process geometry node, betas were less than 10. In addition, it was found that there was beta variation at different emitter current levels for some process types. Remote temperature sensors that compensate for the beta variation of the BJT have circuitry that senses the return current on the base, and adjusts the emitter current to compensate for any variation in beta. This beta compensation ensures that the collector current ratio remains intact. Figure 10 and Figure 11 shows the accuracy before and after beta compensation, respectively. As shown, dramatic improvement in accuracy can be achieved with beta compensation.

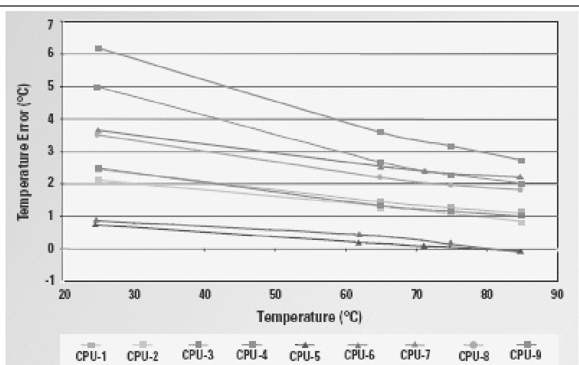


Figure 10. Accuracy Without Beta Variation Compensation

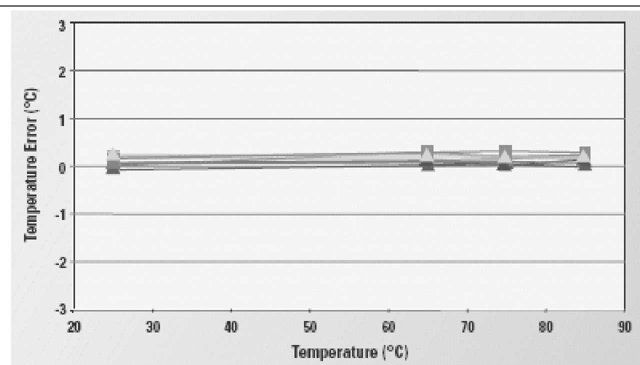


Figure 11. Accuracy With Beta Variation Compensation

Beta variation is simply calculated by measuring the beta of the thermal transistor. Beta is calculated with Equation 12 by forcing different emitter current (I_E) levels and measuring the base current (I_B).

$$\beta = \frac{I_E - I_B}{I_B} \quad (12)$$

Figure 10 shows that beta variation causes offset and slope errors; therefore, beta variation can also be compensated by adjusting offset and n-factor values. The method described in Section 1.2.2 can determine the offset and n-factor settings to adjust for this error. This method can be used with remote temperature sensors (such as the TMP468 device) that do not support beta compensation, but includes offset and n-factor correction registers.

In addition, BJTs in small geometry SOI (Silicon on Insulator) processes do not exhibit beta variation. Discrete transistors have very large beta, so even if beta varies, it does not impact the collector current ratio. Some processor manufacturers include an offset compensation value in memory so that software can access this value and program the remote sensor offset adjust register accordingly.

1.2.4 Series Resistance Cancellation

PCB trace resistance can be an issue if it is too high. Since there will be a voltage drop between the V_{BE} at the transistor and the measured V_{BE} at the temperature sensor, shown in Equation 13, this will introduce a temperature offset. Along with the temperature error, too high of a series resistance can impact performance by causing a voltage at the remote transistor input stage to exceed the design common mode limit. Most remote temperature sensors (such as the TMP468 device) support a limited amount of series resistance cancellation.

$$V_{BE_M} = V_{BE_T} + (R_S \times I_X)$$

- $V_{BE\ T}$ is the Base-Emitter Voltage at the processor or transistor
- $V_{BE\ M}$ is the measured Base-Emitter Voltage at the temperature sensor
- R_s is the series resistance
- I_X is the current being sourced from the temperature sensor (13)

If another current level measurement is used, three in total, the series resistance term can be canceled out by solving the three equations for temperature. Figure 12 shows the input architecture of the TMP468 device. The three current levels are 1 ×, 6 ×, and 16 × (7.5 µA, 45 µA, and 120 µA).

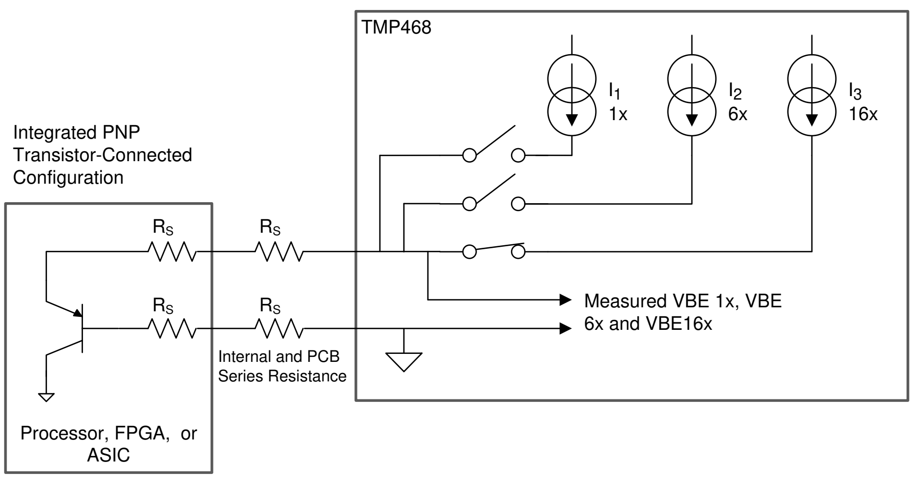


Figure 12. Δ_{VBE} Measurement With Three Currents Cancels Errors Caused by Series Resistance

Using Equation 5 and Equation 13 with three current sources the constants q, k, and n cancel out, both V_{BE} voltages are measured, and all three sourced currents are known, thus we can solve for R_S in Equation 14.

$$\frac{\Delta V_{BE_{M1}} - R_S(I_X - I_Y)}{\ln\left(\frac{I_X}{I_Y}\right)} = \frac{\Delta V_{BE_{M2}} - R_S(I_Y - I_Z)}{\ln\left(\frac{I_Y}{I_Z}\right)} \quad (14)$$

Figure 13 shows the simplified input stage of the TMP468. The area enclosed by the dashed box is repeated for each D+ input. The D– input is common for all the channels. The current waveform (as shown in Figure 13) cycles through the three levels (1 ×, 6 ×, and 16 ×) multiple times during a conversion. The TMP468 device has a ΣΔ ADC architecture that provides good noise immunity. An RC low-pass filter is included that improves the noise immunity.

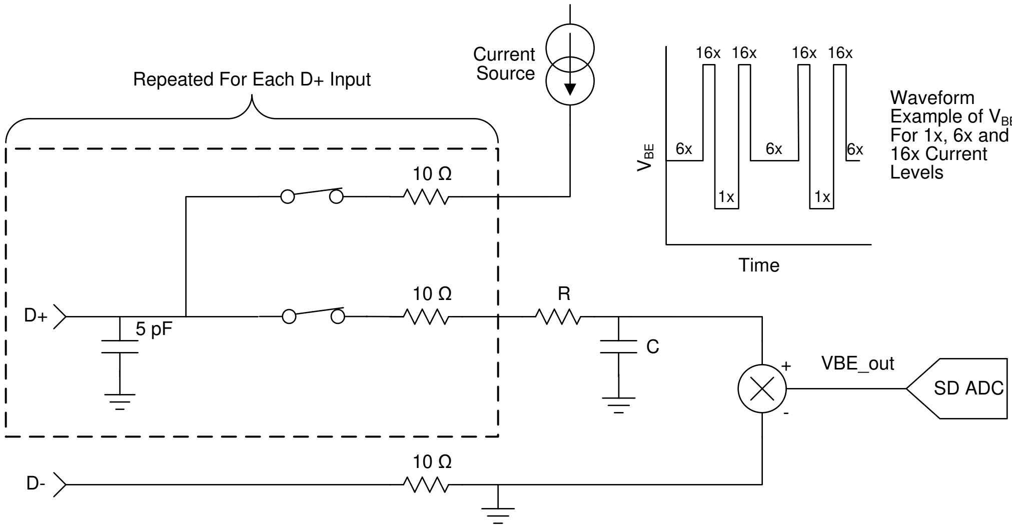


Figure 13. TMP468 Input Stage Simplified Schematic

1.2.5 PCB Leakage Current or Resistance

PCB leakage is another error source that directly impacts the current ratios and causes an error in the temperature reading. Figure 14 shows the actual effect on the temperature reading that is caused by leakage resistance to ground or the power supply voltage. Even the 10-MΩ impedance of an oscilloscope probe can cause several degrees of error. Take care to ensure that the PCB is cleaned properly from fingerprints, flux, and other chemicals that can cause leakage.

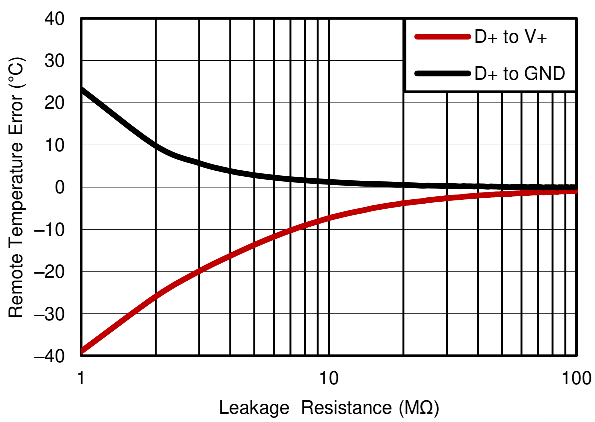
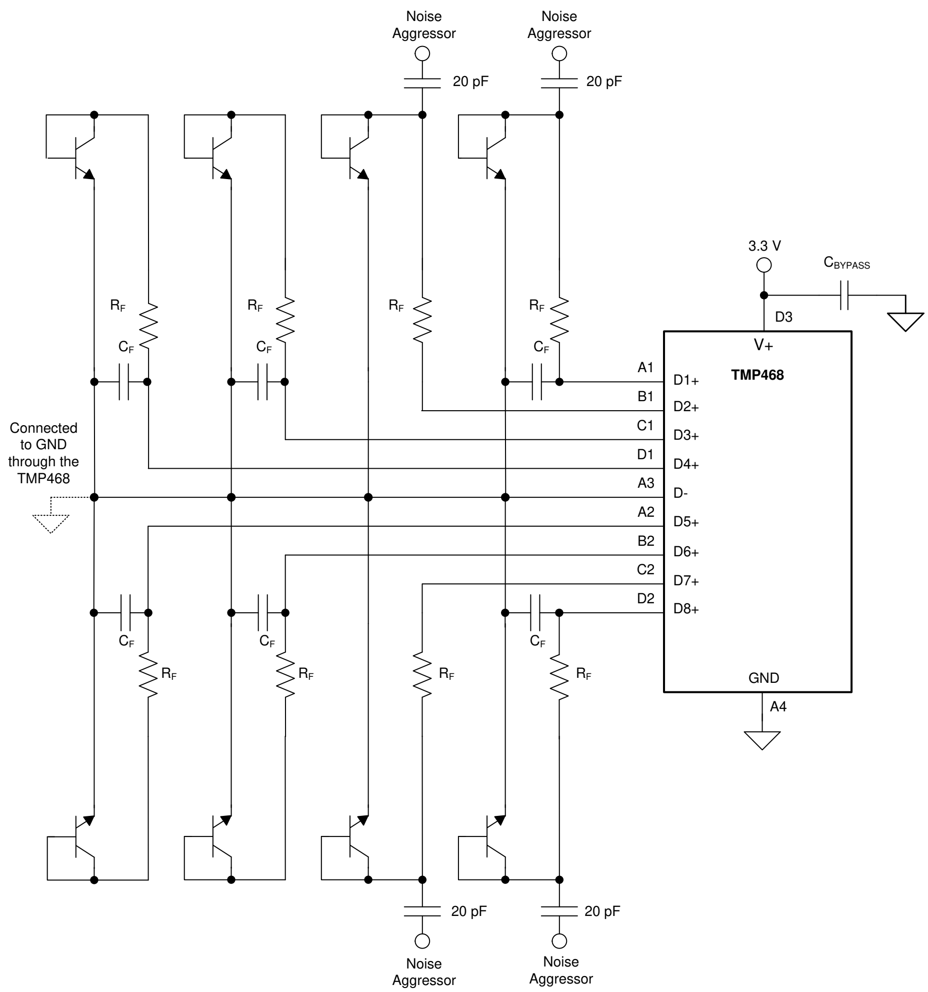


Figure 14. Leakage Resistance Effect on TMP468 Accuracy

1.3 Improving Noise Immunity

Another error source can be caused by EMI or inductive coupling into the remote junction PCB traces. This typically occurs when diode traces run in parallel with high frequency signal lines carrying high currents. Examples of this can be a fast digital clock or placing the remote transistor in close proximity to an inductor of a switching regulator. Without careful PCB layout considerations, the small signal level of the thermal junction voltage Δ_{VBE} of tens of millivolts can be swamped by these error sources. Thus, shielding of traces is required. Inductive coupling is also minimized when trace spacing is greater than 10 mils.

The circuit of Figure 15 was used to determine the effect noise has on the performance of the TMP468 remote temperature sensor. Comparable effects would be expected with other remote temperature sensors. Four channels (D3–D6) were left without any aggressors, but included a filter. The four remaining channels (D1, D2, D7, and D8) included a noise aggressor coupled in through a capacitor. Figure 16 and Figure 17 shows the waveform of the aggressor at two different time bases. This noise aggressor is exaggerated over noise that may be residing in a system.



Note all C_F values are 1 nF and all R_F values are 1 kΩ.

Figure 15. Schematic for Noise Tests

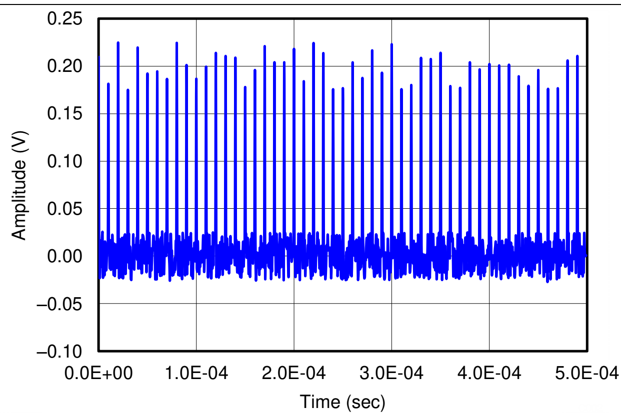


Figure 16. Noise Aggressor Waveform With a 100-µs Time Base

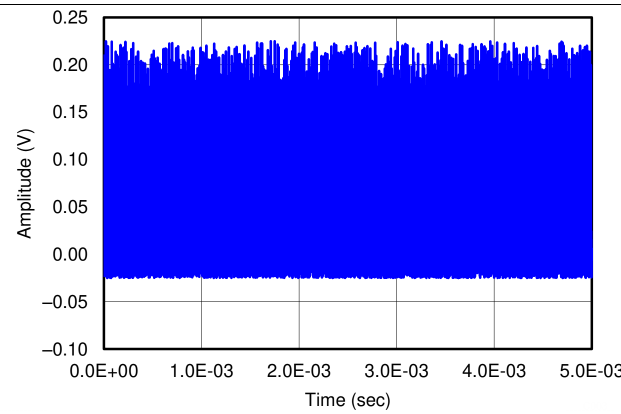


Figure 17. Noise Aggressor Waveform With a 1-ms Time Base

The board residual noise of about 400m°Cp-p can be seen in Figure 18. Notice all channels (including the local temperature overlap) read approximately 21.25°C. The sample rate of the TMP468 was set to once every two seconds. As can be seen, the temperature was very stable for over three minutes. These conditions were also used for the results in Figure 19 where the noise aggressor was injected through 20 pF into channels D1, D2, D7, and D8.

As shown in Figure 19, the channels without the filters (D2 and D7) were impacted severely by the noise aggressor. The channels with the C_F filters (all excluding D2 and D7) were impacted less.

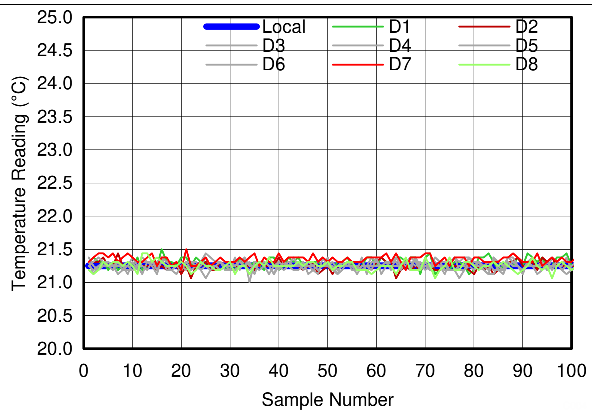


Figure 18. TMP468EVM Baseline Noise Level Measurement Without Noise Injection

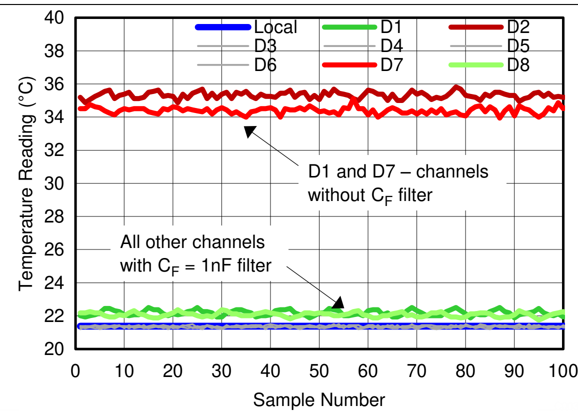


Figure 19. TMP468EVM Noise Level Measurement With Noise Injection

1.4 PCB Layout Considerations

Note in Figure 19 that although channels D2 and D7 have the same filter and noise coupling on the schematic, there is an appreciable difference on the effect of the reading because of different PCB layouts. D2 was run as a single-ended trace without any care as to where the D– trace was placed, as shown in the PCB stack of Figure 20. D7 was run as a true differential pair, as shown in Figure 21.

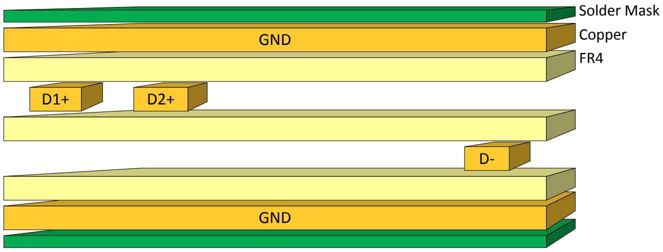


Figure 20. Routing of D+ and D- as Single-Ended Traces on Multiple Layers

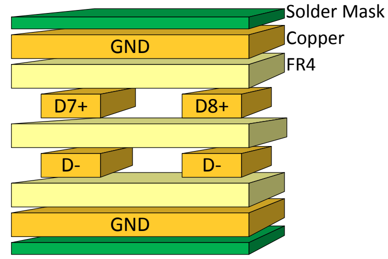


Figure 21. Routing as Differential Pairs on Multiple Layers

The dark red trace in Figure 19 is D2 which used the single-ended layout, while the light red trace shows the response of D7 (the differential pair layout). Figure 19 clearly shows that the differential layout is more effective than the single-ended layout.

2 Troubleshooting a Noisy System

This section describes useful techniques for troubleshooting a noisy board. Use a TMP468 evaluation board to determine where the noise is coming from in the system. Cut the traces between the BJT and the TMP468 and replace the connection with a cable as shown in Figure 22. This determines if the issue is caused by the actual routing on the PCB. Cut the traces to the onboard TMP468 and patch in the TMP468 on an evaluation board into the system as shown in Figure 23. This determines whether the PCB routing to the sensor is suitable, or the issue is caused by a power supply or other source. The EVM software is simple to use with an evaluation board that connects to a USB port on a PC. Headers are provided on most evaluation boards that simplify EVM patching into a system.

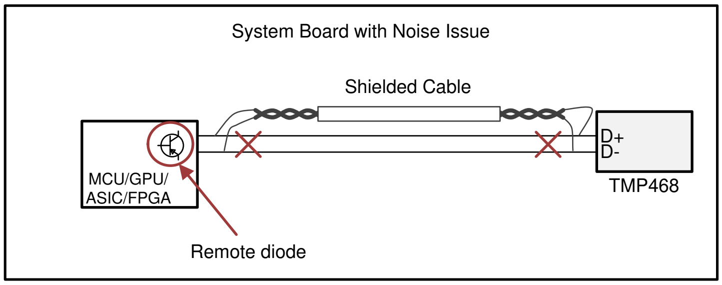
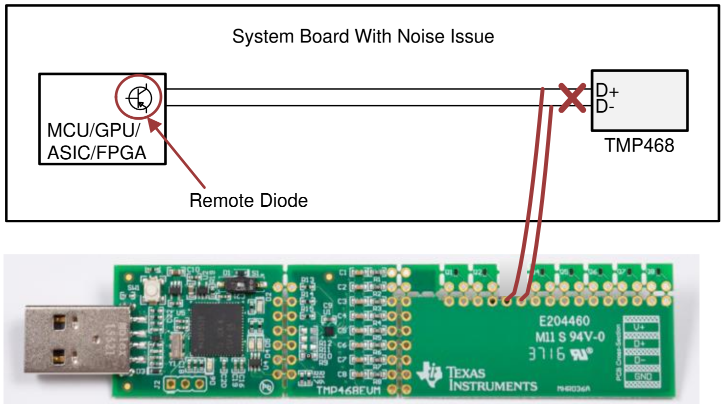


Figure 22. Isolating Noise Coupling Into Remote Transistor Traces



TMP468EVM

Figure 23. Isolating Noise Coupling Through Power Supply or Other Sources

3 Conclusion

There are many design considerations for successfully sensing temperature of a highly integrated system by using a remote temperature sensor. Resistance cancellation helps improve the performance of a remote temperature sensor by enabling filtering and eliminating errors that are caused by series resistance. Layout techniques are also important, with the differential pair routing of the D+ and D– lines resulting in the highest performance. N-factor and offset adjust registers help overcome process variations with different BJT manufacturers. All these techniques assist in building a robust system with remote temperature sensors.

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Original (January 2017) to A Revision **Page**

- Changed application report title 1
- Removed TMP411 references from the document. Added references to the TMP431 1
- Changed diode sensor references to temperature sensors for clarity 1
- Changed 'hundreds of microvolts' text to 'tens of millivolts' for clarity 6
- Changed *Compensating for Errors Using Lab Measurements* section 8
- Changed *Series Resistance Cancellation* section 11