AN2017

PSoC® 1 Temperature Measurement With Thermistor

Author: David Van Ess
Associated Project: Yes
Associated Part Family: CY8C28xxx
Software Version: PSoC® Designer™ 5.4
Related Application Notes: AN66477

If you have a question, or need help with this application note, contact the author at msur@cypress.com.

AN2017 shows how to use PSoC® 1 to accurately measure temperature with a thermistor. The associated project measures the resistance of a thermistor to calculate its temperature using lookup tables and equations, and is also used with other PSoC 1 devices that have the required resources.

Introduction

A thermistor is a temperature-sensitive resistor in which resistance varies with temperature. There are two types of thermistors: positive temperature coefficient (PTC) thermistors and negative temperature coefficient (NTC) thermistors. This application note describes the more commonly used NTC thermistors, in which resistance decreases with increase in temperature. Based on this principle, temperature is calculated by measuring the resistance. Thermistors are available as elements, probes, and in packages designed for specific end applications. Their resistances can vary typically from a few Ω to several kΩ.

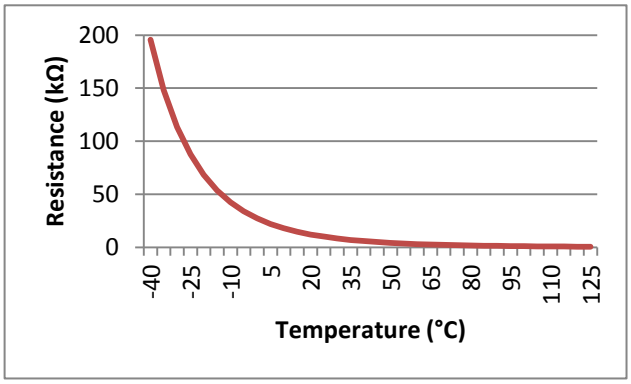
This application note comes with an associated project, which measures the resistance of the thermistor. The temperature is then calculated using an equation or a lookup table. This application note also comes with an excel sheet that calculates the Steinhart-Hart constants and builds the lookup table. This application uses the following PSoC 1 resources:

- Essential
  - ACDINCVR User Module
  - PGA User Module
  - Two analog output buffers
  - Analog Multiplexer
- Optional
  - Display (LCD)

Thermistors: A Primer

The variation of resistance with temperature for a thermistor is nonlinear. Figure 1 shows a typical resistance versus temperature of a thermistor.

Figure 1. Resistance versus Temperature Curve of Thermistor NCP18XH103F03RB



As explained earlier, a NTC thermistor is a semiconductor device that becomes less resistive as its temperature increases. The change in resistance is roughly expressed by the following equation.

$$\frac{R(t1)}{R(t2)} = A^{(t1-t2)}$$ Equation 1

Where:

A is an empirical constant less than one.

t1 and **t2** are two different temperatures.

R(t1) and **R(t1)** are the resistances at these temperatures.

"Roughly," in this case, means that it is a great equation for some academic introduction to semiconductor materials, but will not do for any real world, temperature-measuring application.

The Steinhart-Hart equation describes the resistance change of a semiconductor thermistor as related to its temperature. The following equation shows it to be a third-order logarithmic polynomial using three constants.

$$\frac{1}{T_K} = A + B \cdot \ln(R) + C \cdot \ln(R)^3 \qquad \text{Equation 2}$$

Where:

A, **B**, and **C** are empirical constants.

R is the thermistor's resistance in Ω.

T_K is the temperature in kelvin.

A more useful equation shows the temperature in Celsius.

$$T_C = \frac{1}{A + B \cdot \ln(R) + C \cdot \ln(R)^3} - 273.15 \qquad \text{Equation 3}$$

Temperature calculations are only as accurate as the resistance measurement of the thermistor.

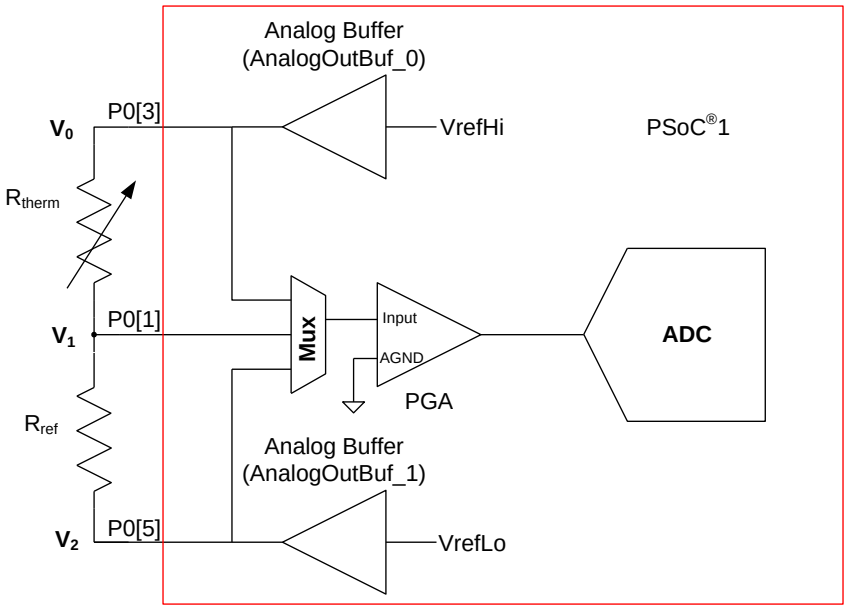
Some datasheets provide the three Steinhart coefficients (A, B, and C). Other datasheets provide "Temperature coefficient" (Alpha) values, "Sensitive index" (Beta) values, or both. Although the Alpha or Beta coefficients can determine temperature, they are limited to a specific temperature range for which they are specified. The Steinhart-Hart equation does not have this limitation.

Because the parameters provided for thermistors can vary, their usage and interchangeability in an application can be complicated. To address this issue, the attached *AN2017_S_H_Constant_Calc.xls* file, calculates the required A, B, C Steinhart-Hart coefficients, based on the resistance versus temperature table or curve available in datasheets. If the resistance versus temperature value is not provided in the datasheet, users can measure them on a test bench.

Reading Ohms the PSoC Way

The setup to measure the resistance of a thermistor using PSoC 1 is shown in Figure 2. PSoC 1 Output Buffers and Input Multiplexer are connected to significantly remove gain and offset errors from the resistance calculation.

Figure 2. Measuring Ohms the PSoC Way



The current through R_{ref} also flows through the thermistor, which give us:

$$\frac{V_0 - V_1}{R_{therm}} = \frac{V_1 - V_2}{R_{ref}} \qquad \text{Equation 4}$$

Solving for $R_{thermistor}$, we get the following equation:

$$R_{therm} = R_{ref} * \left(\frac{V_0 - V_1}{V_1 - V_2} \right) \qquad \text{Equation 5}$$

As shown in the previous equation, any offset errors in the measurement system are removed by the subtraction of two measured voltages. The ratio of these two different values removes any measurement path gain error. This leaves the measurement error to be determined by R_{ref}. The reference resistor's accuracy requirement is determined by the specific application requirements.

This is valid as long as the measured signal is never outside the range of the ADC. To guarantee this the PGA is set for a gain slightly less than unity.

Interface With PSoC 1

Figure 2 shows that there are only two discrete components required outside the PSoC 1:

- The thermistor
- The reference resistor

Thermistor

For this application, a NCP18XH103F03RB thermistor is selected. This thermistor is available on the CY8CKIT-025 Temperature Sensor EBK. It has the following specifications:

- 10,000 Ω at 25 °C
- –40 °C to +125 °C operating range

Calculate the Steinhart-Hart Constants for the Thermistor using the following equation.

$$\frac{1}{T_K} = A + B \cdot \ln(R) + C \cdot \ln(R)^3 \qquad \text{Equation 6}$$

To use the Equation 6, to calculate the temperature with the measured resistance, the three Steinhart-Hart constants A, B, C are required. To solve for the three constants we need three equations. Table 1 gives the resistance of the thermistor at three different temperatures. These values have been taken from the device's datasheet.

Table 1. Three Data Points for NCP18XH103F03RB Thermistor

Temperature (°C)	Resistance (Ω)
0	27219
25	10,000
80	1669

Feed the data from Table 1 in the appropriate location in the attached excel sheet AN2017_S_H_Constant_Calc.xls

- Temperature in Red Blocks
- Resistance in Green Blocks

The Steinhart-Hart constants are calculated and shown in the Blue blocks

The generated lookup table is shown in Purple.

Note:

- The lookup table is generated for the temperature range 0 °C to 80 °C in steps of 1 °C.
- Smaller the difference between upper and lower bound, more reliable will be the values of the Steinhart-Hart constants generated.

Table 2. Steinhart-Hart Coefficients for NCP18XH103F03RB

Steinhart-Hart Coefficient	Value
A	0.000891358
B	0.000250618
C	0.000000197

Reference Resistor

Given an ADC of finite resolution, the most accurate measure is made when:

$$R_{thermistor} = R_{ref} \qquad \text{Equation 7}$$

When the equation holds true, each resistor has half of the ADC's range across it. Half of the range effectively cuts the resolution by one bit. If one resistance becomes four times bigger than the other, then 80% of the range is across the larger resistance and 20% across the smaller. 80% of the range is effectively a reduction of resolution of one third of a bit. A 20% range reduces the resolution by over two bits.

The problem is that the thermistor resistance, over temperature varies several decades in magnitude. Table 1 verifies this.

For this case, a reference resistor of 10 kΩ is selected for the most resolution at 25 °C. With the ADC set for 13 bits, the resolution of the reference resistor and thermistor at three different temperatures is shown in Table 3:

Table 3. Effective Resolution for a 13-Bit ADC

°C	$R_{thermistor}$ Ohms	$R_{thermistor}$ ADC Resolution	$R_{reference}$ Ohms	$R_{reference}$ ADC Resolution
0	27219	13 Bits	10,000	8 Bits
25	10,000	12 Bits	10,000	12 Bits
80	1669	8 Bits	10,000	13 Bits

With a 1% tolerance in thermistor resistance, eight bits of resolution is adequate.

For this example, the architecture in Figure 1 is used. The reference resistor is selected to be 10 kΩ. Because the thermistor has an uncertainty of 1%, choosing a tolerance of 0.1% for the reference resistor removes any error it can contribute.

Sample Project

Figure 3 shows the User Module placement, following the schematic in Figure 1.

PGA User Module - Buffer

The Buffer is a PGA User Module placed in ACB00. It is connected to the multiplexer AMUX4, which connects to external pins.

Software enables ACC00's testmux to connect $V_{refHigh}$ to the column 0 analog bus. Select AnalogOutBuf_0 in the Interconnect View to bring this reference out to P0[3] as shown in Figure 3.

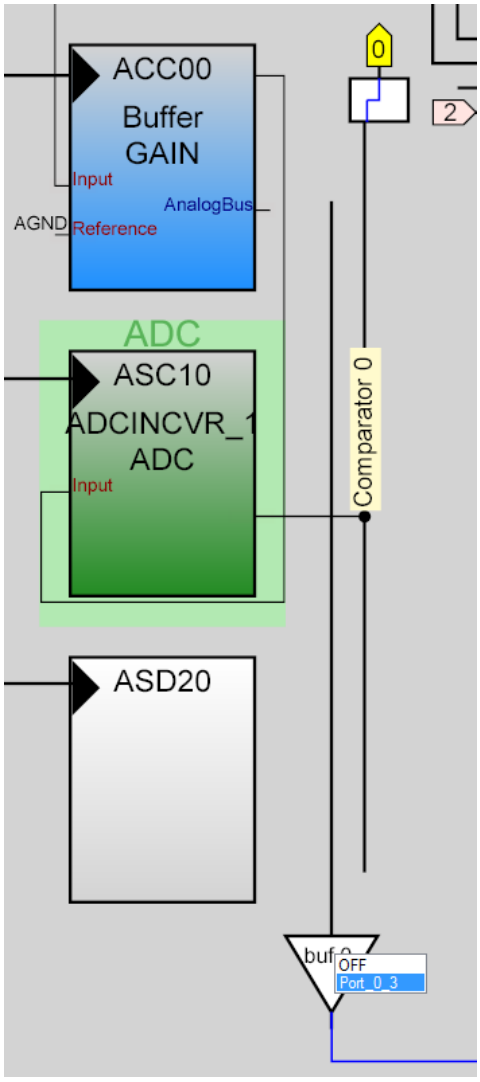
PGA User Module - RefLow

A second PGA User Module in ACC01 as a placeholder. The gain stage is not used. The sole purpose is to allow access to the testmux. Software enables ACC01's testmux to connect V_{refLow} to the column 1 analog bus. Select AnalogOutBuf_1 in the Interconnect View to bring this reference out to P0[5] (similar to the Figure 3).

ADCINCVR User Module

The analog block of the ADCINCVR is placed just below the PGA User Module - Buffer, from which it receives the signal. The clock for the ADCINCVR, with a 13-bit resolution, is set to 333 kHz for a sample rate of 10 sps. This sample rate causes any 60 Hz or 50 Hz interference to be removed from the signal (sampling at a sub-multiple of a frequency will reject that frequency). The sample rate can be increased if the application requires a faster conversion.

Figure 3. Routing Analog Out Buffer to a Pin



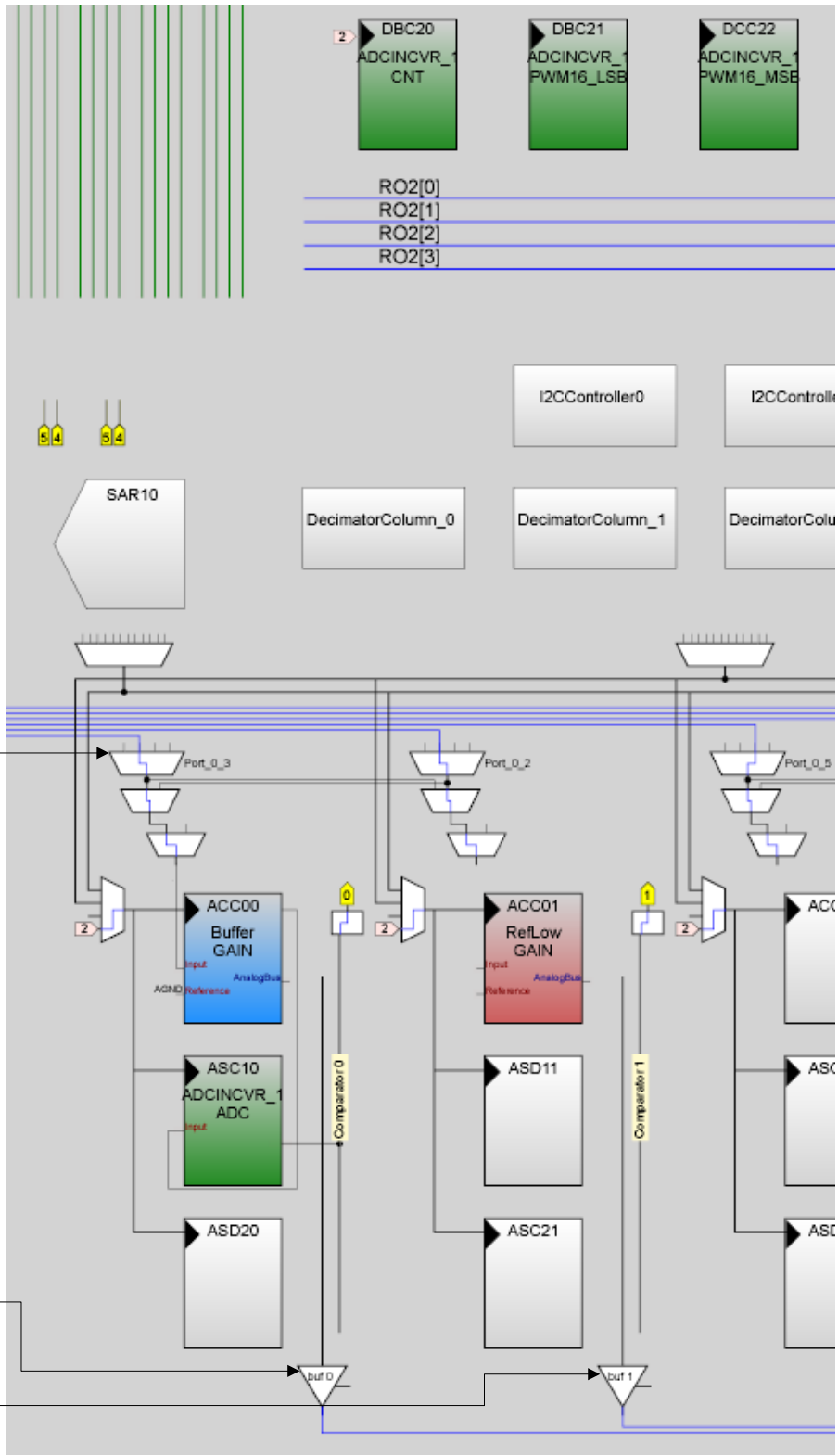
AMUX4 User Module – AMUX4

AMUX4 User Module is placed at AnalogColumn_InputMux_0. AMUX4 is used to switch between the three pins P0[1], P0[3], and P0[5].

LCD User Module

LCD is used to display the calculated temperature and resistance of the thermistor.

Figure 4. User Module Placement



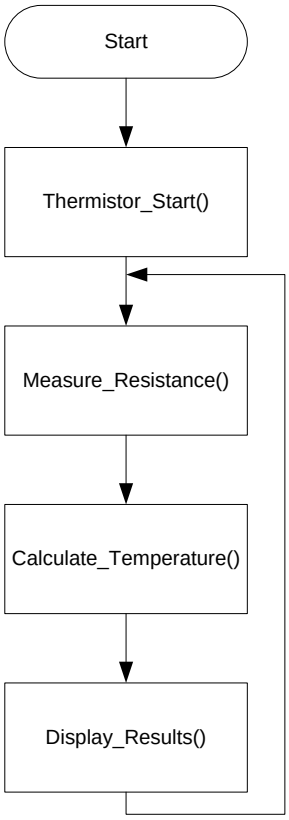
AMUX4
(AnalogColumn_InputMux_0)

Analog Output Buffer
(AnalogOutBuf_0)

Analog Output Buffer
(AnalogOutBuf_1)

Firmware

The firmware is written in C and is explained in this section.



1. Thermistor_Start()
   a. Performs the required initialization for the User Modules involved.
   b. Enables ACC00's testmux to connect $V_{refHigh}$ to the column 0 analog bus.
   c. Enables ACC01's testmux to connect V_{refLow} to the column 1 analog bus.
2. Measure_Resistance()
   a. This function as stated measures the resistance of the thermistor.
   b. The voltage at P0[3], P0[1] and P0[5] is measured.
   c. The resistance value is calculated and stored to be processed further. The value is calculated by both long or float math using Equation 5.
3. Calculate_Temperature()

   After measuring the thermistor resistance, it must be converted to a temperature value. This can be done either with float or long math:

   - Float Math
     - Plug the thermistor constants into the Steinhart-Hart equation to calculate the temperature. This has the advantage of being the most accurate. Its disadvantage is that it requires floating-point math. The Steinhart-Hart coefficients are calculated in the attached *AN2017_S_H_Constant_Calc.xls* file.
   - Long Math
     - Using the Steinhart-Hart coefficients, calculate a table of temperature versus resistance over the range required (This is done in the attached *AN2017_S_H_Constant_Calc.xls* file). This table can be in line integers. Finer resolution can be obtained if required. This has the advantage of being faster to calculate. The disadvantage is the ROM space used to store the table and is less accurate.

   Both techniques are valid; it is up to the user to decide which best fits his or her application. For this example, both methods are implemented. The user has to comment out the line of code in the header file *thermistor.h* to select the method of measurement.

   ```
   #define LONG_MATH
   or
   #define FLOAT_MATH
   ```

4. Display_Results()
   a. The measured temperature and resistance of the thermistor are displayed on the LCD.
   b. This function can be updated if the user wants to procure data through another interface such as UART/I^2C.

Add the files *thermistor.c* and *thermistor.h* to the project. Copy the code given below into *main.c*.

```
#include "m8c.h"
#include "PSoCAPI.h"
#include "thermistor.h"

void main(void)
{
        M8C_EnableGInt;

        Thermistor_Start();

        while(1)
        {
                Measure_Resistance();
                Calculate_Temperature();
                Display_Results();
        }
}
```

Note Update the Steinhart-Hart coefficients and the LUT (LUT is needed only if LONG_MATH is used for calculations) from the attached file, *AN2017_S_H_Constant_Calc.xls*, which was updated as explained in section Thermistor.

A comparison of the two methods for measurement and calculation is given in Table 4.

Table 4. Comparison of Long and Float Methods

Method	Time Taken	ROM Used
Long	310 ms	3779 kB
Float	340 ms	7966 kB

Evaluate the Example Project

1. Build the associated project for the required type of method to be used i.e., Long_Math or Float_Math. (Refer the section Firmware).
2. Program the CY8CKit-001.
3. Connect the CY8CKit-025 to CY8CKit-001 as shown in the Table 4.
4. Power the board and observe the results on the LCD.

Thermistor User Module

PSoC Designer 5.4 includes a thermistor user module in the user module catalog (see Figure 6), which implements the 'Long-math' method explained in the application note's example project for temperature measurement using a thermistor.

Hardware Connection

Table 5 lists the hardware connections between the two boards:

Table 5. Hardware Connections

	Wire	CY8CKit-001	CY8CKit-025
RefHi	Red	P0[3]	P0[0]
Signal	Yellow	P0[1]	P0[1]
RefLow	Black	P0[5]	GND_A
LCD	N/A	Port_2	N/A

Figure 5. CY8CKit-025 Connected to CY8CKit-001

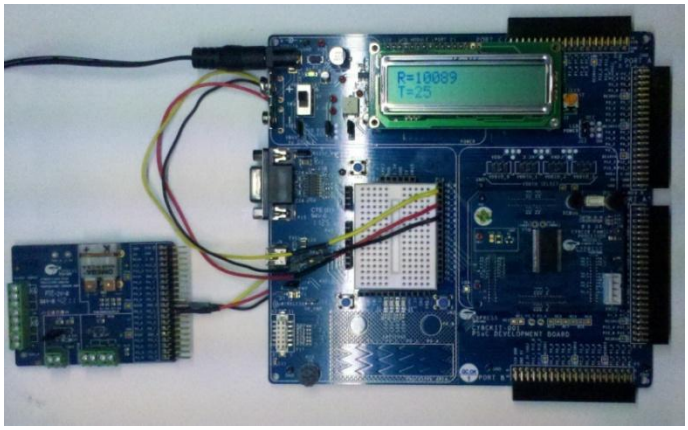
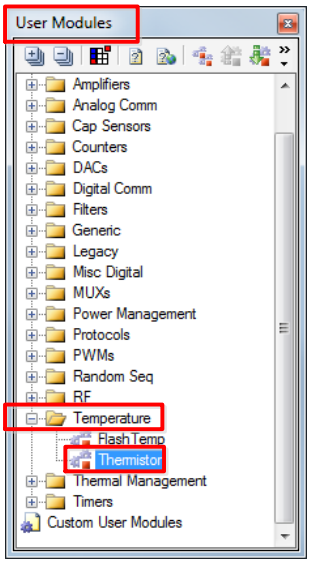
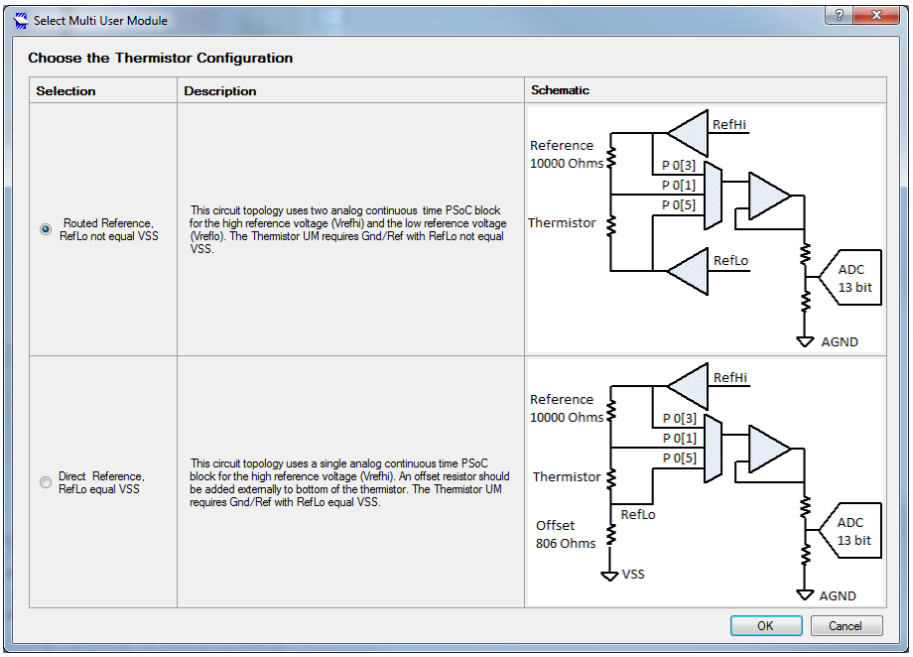


Figure 6. Thermistor User module in the Catalog



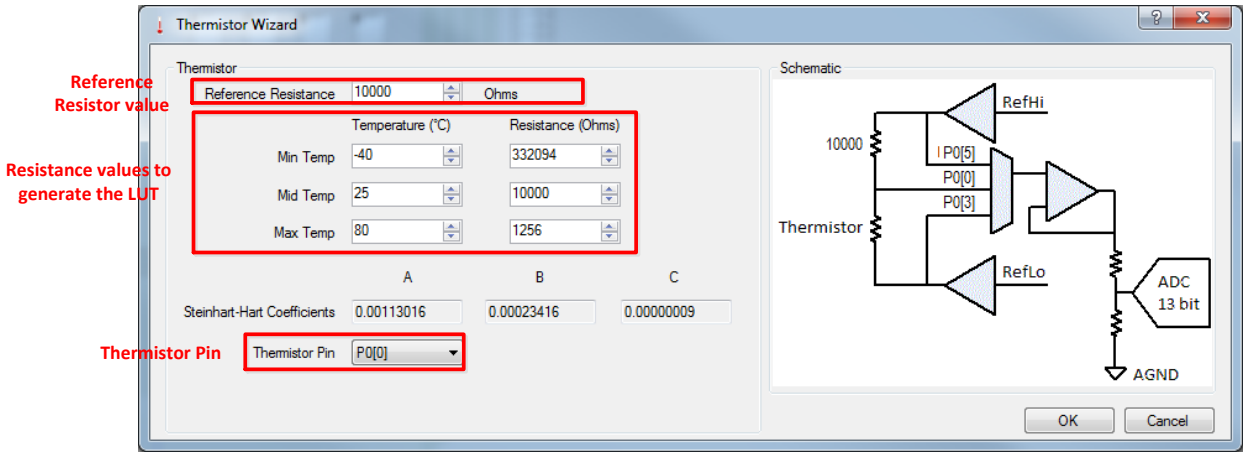
The user module provides the user option to have either VSS as RefLo signal or RefMux's RefLo signal as the RefLo for the thermistor connection (see Figure 7).

Figure 7. Thermistor User Module RefLo Selection



The user module wizard generates the LUT automatically without your entering the values in the *AN2017_S_H_Constant_Calc.xls* file and copying the values in to the code. You need to enter resistance values at three temperature settings – Min, Mid, and Max temperature setting – in the user module wizard as shown in Figure 8.

Figure 8. Thermistor User Module Wizard

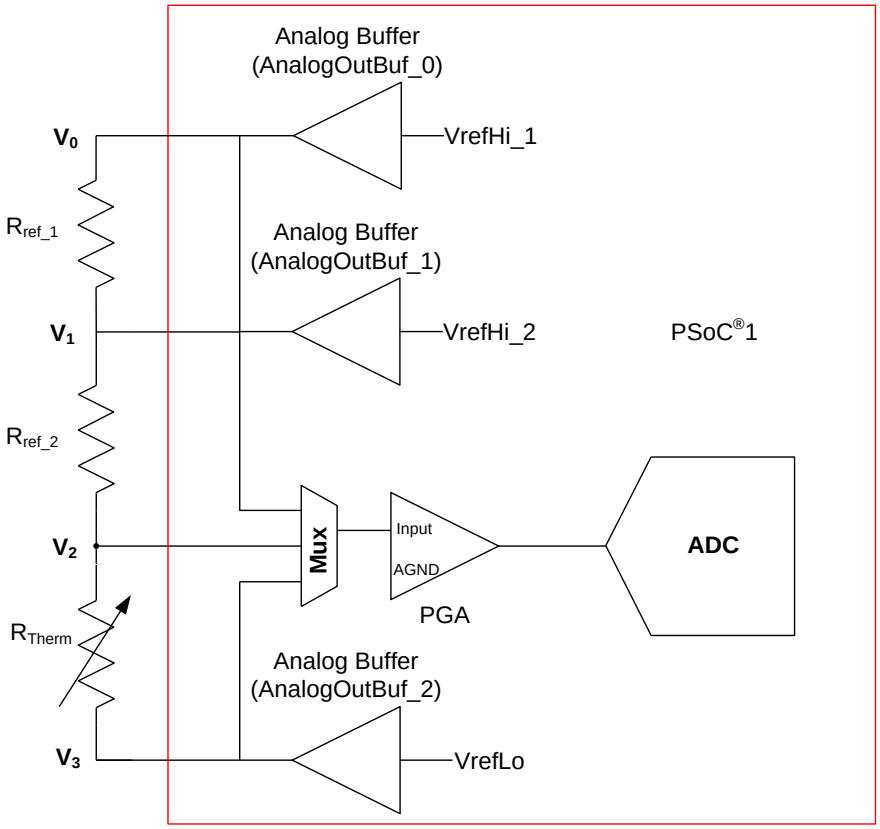


For details on the user module and its usage, refer to the Thermistor user module datasheet.

How to Improve the Resolution?

More resolution can be obtained with use of multiple reference resistors. Figure 3 shows architecture to allow multiple reference resistors.

Figure 9. Selectable Reference Resistors



When AnalogOutBuf_0 is disabled and AnalogOutBuf_1 is driven, the reference resistance is R_{ref_1}. V_1 is sensed through R_{ref_2}. When AnalogOutBuf_1 is disabled and AnalogOutBuf_0 is driven, the reference resistance is $R_{ref\_1} + R_{ref\_2}$.

Although this architecture allows better resolution, it does so at the cost of an external pin, a resistor, and an extra buffer (and its power).

Self-Heating of Thermistor

Self-heating is a phenomenon in which the thermistor temperature increases because of the current flow through it. This self-heating introduces an error in the measured temperature. The self-heating effect is provided as dissipation factor (mW/°C) in the datasheet. It is defined as the power required to raise the temperature of the thermistor by 1 °C above the ambient temperature and is expressed as Equation 6.

$$\text{Dissipation Factor} = T_{Power} / T_{error} \quad \text{Equation 8}$$

Where T_{Power} is the power supplied to the thermistor and T_{error} is the difference between the ambient and the measured temperature value.

The NCP18XH103F03RB thermistor, considered as an example in this application note, has a dissipation factor of 1 mW/°C. Consider voltages supplied for V_0 and V_{ss}, shown in Figure 1, are 3.9 V and 1.3 V. Solving Equation 6 for a thermistor with resistance of 10 kΩ at 25 °C, the temperature error (T_{error}) is 0.17 °C. This error can be decreased by decreasing the reference voltage range, and thus the current flow through the thermistor.

Summary

The right circuit topology makes it possible to measure a resistance with its accuracy determined by a single reference resistor. An understanding of the Steinhart-Hart equation makes conversion to temperature, either by calculation or table lookup, a straightforward task.

Document History

Document Title: AN2017 - PSoC® 1 Temperature Measurement With Thermistor

Document Number: 001-40882

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	1536344	JVY	10/07/2007	OLD APP. NOTE: Obtain spec. # for note to be added to spec. system.
*A	3155592	YARA	01/27/2011	Changed the title. Added Calculations section. Updated the content.
*B	3260418	YARA	06/14/2011	Added a section on self-heating.
*C	3692836	ADIY	07/26/2012	Created and excel file to calculate Steinhart-Hart constants and the Lookup Table. Updated for Thermistor NCP18XH103F03RB on CY8CKit-025. Updated associated project to CY28xxx. Updated firmware to be more modular. Removed reference to obsolete Application Notes. Updated template.
*D	4323179	MSUR	03/27/2014	Added section on Thermistor User module Updated project to PSoC Designer 5.4
*E	5713448	AESATMP9	04/26/2017	Updated logo and copyright.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709