

Creating a Polynomial for TMP6 Temperature Measurements



Gordon Varney

ABSTRACT

The most basic method of getting a temperature from an NTC is by using a look-up table. A look-up table; however, is typically in 1°C or 5°C data steps. When a look-up function is performed, the temperature data is in the 1°C or 5°C steps based on the look-up table. To get better resolution, perform an interpolation on the look-up results. The interpolation is a linear interpolation process so the results between the upper and lower temperature selected from the look-up table are not part of the natural curve found in an NTC, but rather a linear step function between two points in the look-up table.

An alternate method of getting a temperature value is to use a Steinhart-Hart equation. The Steinhart-Hart equation is a model of the resistance of an NTC at three different temperatures in the selected temperature range. The Steinhart-Hart coefficients, vary depending on the manufacture and part number of the thermistor and the temperature range of interest. This method requires the use of natural log functions to follow the non-linear slope of the NTC. It is necessary to calculate the coefficients one time, then these coefficients are used in the formula that is run in the controller to get the actual temperature from the NTC. Enter the resistance value in the controller that is calculated from the ADC bit value. In the Steinhart-Hart equation use the calculated three coefficients and run the formula using the calculated resistance. This method is dependable and is more accurate than a look-up function with interpolation. Because the Steinhart-Hart equation uses natural logs it takes a lot of processor time and system resources to calculate.

Enter new technology. Texas Instruments has a new silicon thermistor called the TMP6. This device has a linear PTC (Positive Temperature Coefficient) slope. Most engineers think of a thermistor as a resistor. Since the TMP6 series of parts are based on a CMOS Deep N-Well resistor, think of it as a resistor but it is silicon.

The resistance of the TMP6 parts cannot be measured directly with a multimeter. To get the resistance of the TMP6, it must be calculated from the ADC voltage. TI provides a resistance look-up table to make it similar to an NTC. However, the resistance is calculated from the voltage slope not the other way around. These parts are more accurate than an NTC if TI's guidelines are followed.

While most thermistors use resistance look-up tables, the TMP6 family make it possible to go from the voltage that is measured by the ADC directly to the temperature and disregard resistance altogether. When using the voltage to get the temperature the math is easier and more accurate. Fewer calculations by not calculating voltage drops and currents make using a polynomial equation faster and very accurate. The linear slope of this device allows for a different type of math that can model the curve of the slope very accurately using a simple 4th-order polynomial. This may sound complicated but a 4th-order polynomial is easy to create with the help of Microsoft® Excel® and easy to calculate using addition and multiplication. No natural logs, no look-up table, and no interpolation is needed. The temperature is available directly from the voltage of the ADC.

Table of Contents

1 Capturing the Data.....	3
2 Creating the Plot for the Fourth-Order Polynomial.....	4
3 Creating the Scatter Plot.....	5
4 Creating a Polynomial for the Collected Data.....	6
5 C Code.....	8
6 Creating a Polynomial for an NTC.....	8
7 Summary.....	9

List of Figures

Figure 1-1. Microsoft® Excel® Data Collected.....	3
Figure 2-1. Plot Relationship for the Polynomial.....	4
Figure 3-1. Creating a Scatter Plot.....	5
Figure 4-1. Polynomial Flow.....	6
Figure 4-2. Format the Trendline Label.....	7
Figure 5-1. C Code.....	8
Figure 6-1. NTC Polynomial Curve Fit.....	8

Trademarks

Microsoft® and Excel® are registered trademarks of Microsoft Corporation.
All trademarks are the property of their respective owners.

1 Capturing the Data

To capture the data needed to create the polynomials, mount the desired part onto a test board. The test board can be an EVM or any UUT (Unit Under Test) board. A data logging method is required to capture the ADC voltage at each preferred specific step. An example is a desired temperature range from -40°C to 125°C . Place the UUT inside a temperature chamber that is capable of sweeping from -40°C to 125°C . A moving sweep is a slow sweep that is always moving in the direction selected. The moving sweep must be programmed with a slope. The temperature slope must be between 2–3 minutes per degrees C, or about 330 minutes for a full sweep. In a temperature chamber enough time must be allowed for the entire circuit board to saturate at a given temperature. A smaller board becomes saturated quickly and is brought to temperature quicker; therefore, the test is faster. If capturing data at 1°C degrees steps, then the temperature must be captured every few seconds to line up the reference temperature with the UUT ADC voltage at that temperature. This usually requires a high-end chamber controller.

The simple method is to set the chamber to the desired temperature and allow time for the UUT to come up to temperature or saturate the UUT. Next, record the ADC voltage at that temperature. Set the temperature for the next data point and repeat the process until the entire range of data points is captured. The standard is 1 C degree steps for an NTC for a total of 165 steps. However, the TI TMP6 parts are very linear across the temperature range, so 5°C degree steps work very well for these parts. With a 5°C degree step, 33 data points are captured at -40°C , -35°C , -30°C , through 115°C , 120°C and 125°C . Align the voltage and temperature in a Microsoft Excel spreadsheet in the order as demonstrated in [Figure 1-1](#). The voltage is on the left and the temperature is on the right. The importance of this setup becomes obvious later in the process.

Div Voltage	Temp
1.31206023	-40
1.33713169	-35
1.3624979	-30
1.38809869	-25
1.41387829	-20
1.43978528	-15
1.46577238	-10
1.49179635	-5
1.51781779	0
1.54380094	5
1.56971352	10
1.5955265	15
1.6212139	20
1.6467526	25
1.67212213	30
1.69730449	35
1.72228392	40
1.7470468	45
1.77158138	50
1.79587767	55
1.81992729	60
1.84372327	65
1.86725995	70
1.89053287	75
1.91353858	80
1.93627458	85
1.9587392	90
1.98093151	95
2.0028512	100
2.02449853	105
2.04587426	110
2.06697953	115
2.08781585	120
2.10838503	125

Figure 1-1. Microsoft® Excel® Data Collected

2 Creating the Plot for the Fourth-Order Polynomial

The plot starts with choosing the correct chart type for the data. Selecting the wrong plot type can easily happen when it comes to line charts and scatter plots. They look very similar, especially when a scatter plot is displayed with connecting lines. There is a big difference in the way each of these chart types present data along the horizontal and vertical axes.

In a line chart, the values are displayed as two separate data points that are evenly distributed along the horizontal axis. This is because in a line chart, the vertical axis is a value axis and the horizontal axis is a category axis. Instead of displaying values, a category axis shows evenly spaced groupings (categories) of data. Because the data has only values and no categories, Microsoft Excel automatically generates numbered categories along the category axis, using numbers 1 through 9 on the x axis. The category axis can be changed and a different number set can be used on the x axis, but it is still a category grouping.

Conversely, a scatter chart has two value axes – it displays values rather than categories on the horizontal axis. Therefore, a scatter plot displays the data values as x values on the horizontal axis, and the chosen reference values as y values on the vertical axis. Because this chart type has two value axes, it can display a single data point at the intersection of each x and y value.

The first column selected is the x axis, while the second data column is the y axis. The goal here is to know what ADC voltage value is needed for each temperature value. Displaying the temperature (y axis) and intersecting the voltage (x axis) in [Figure 2-1](#) allows calculation of the temperature from the known voltage using a polynomial equation based on the curve of the plot.

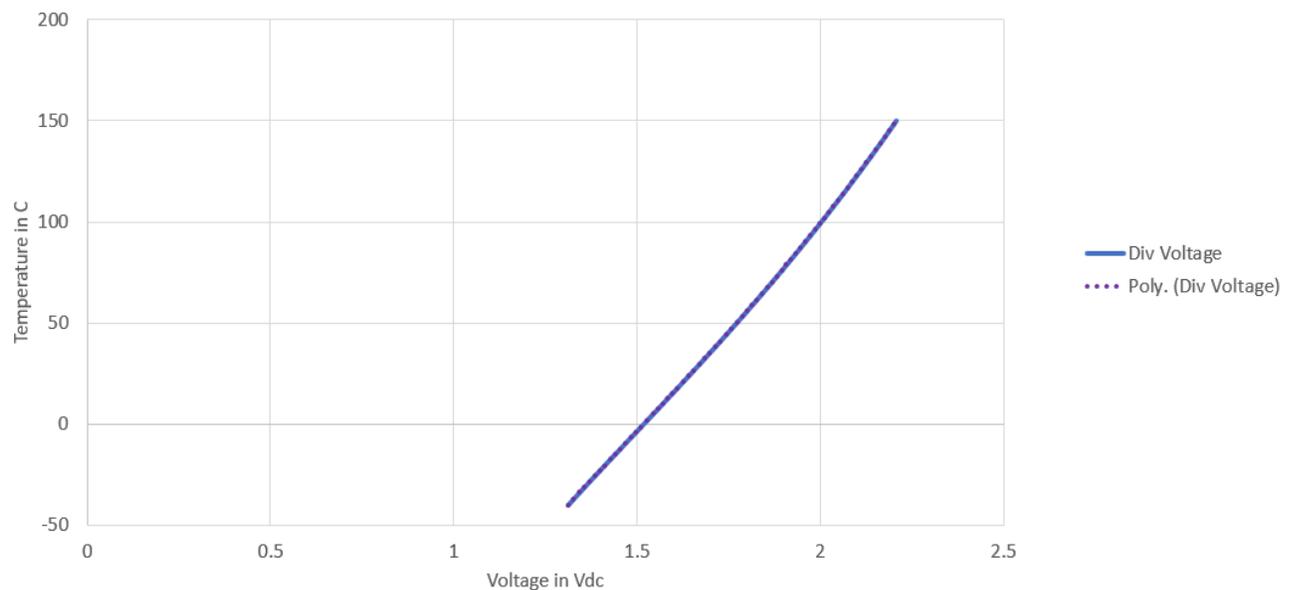


Figure 2-1. Plot Relationship for the Polynomial

3 Creating the Scatter Plot

This section shows how to create the scatter plot using Microsoft Excel.

1. Highlight the created data. Make sure the voltage (x axis) is on the left column and the temperature (y axis) is in the right column.
2. Select *Insert* in the top menu bar. Find *Scatter* plots in the *Charts* section. Select scatter plot with smooth lines as the desired plot type.

These steps yield a plot similar to [Figure 3-1](#) with the temperature data on the left in the y axis and the voltage data on the bottom in the x axis.

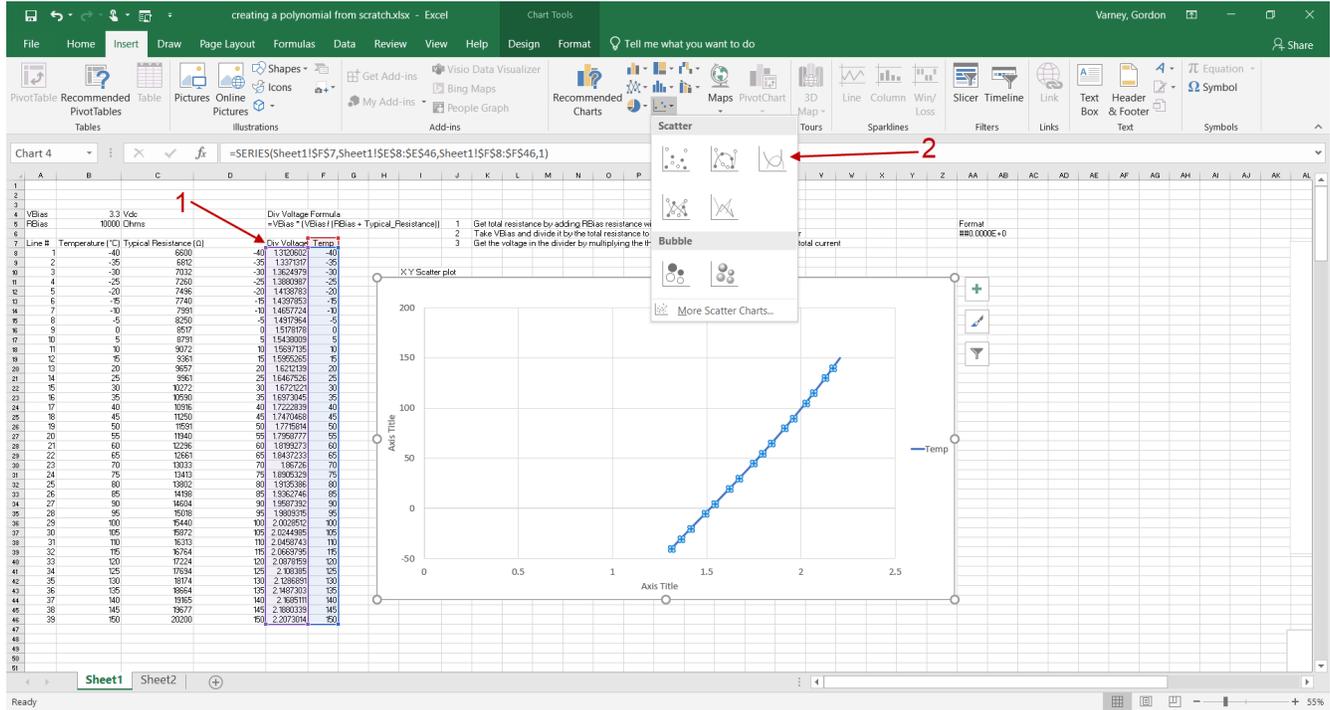


Figure 3-1. Creating a Scatter Plot

4 Creating a Polynomial for the Collected Data

Polynomial curve fitting is the process of constructing a curve or mathematical function that has the best fit to a series of data points. In mathematics, a polynomial is an expression consisting of indeterminates and coefficients, that involves only the operations of addition, subtraction, multiplication, and non-negative integer exponentiation of variables. In theory, the polynomial coefficients are always positive and the function has operations of addition, subtraction, and multiplication. The goal in programming is to fix the code and only have to change out the coefficients. A polynomial equation is modified in code to only use addition and multiplication so that the formula remains constant. So positive and negative coefficients are used to perform the addition and subtraction needed. For the TMP6 parts, a 4th-order polynomial is used to get the best curve fit for the near linear curve of the PTC thermistor.

Use the following steps to create the polynomial from the Microsoft Excel scatter plot previously created. Follow the flow in [Figure 4-1](#) for each step of the process.

1. Right click the plot line
2. Select *Add Trendline*, then the trendline must be formatted
3. Select the *Polynomial* trendline option
4. Change the *Order* to "4", for 4th-order
5. Check the box for *Display Equation on chart*
6. Check the box for *Display R-squared value on chart*

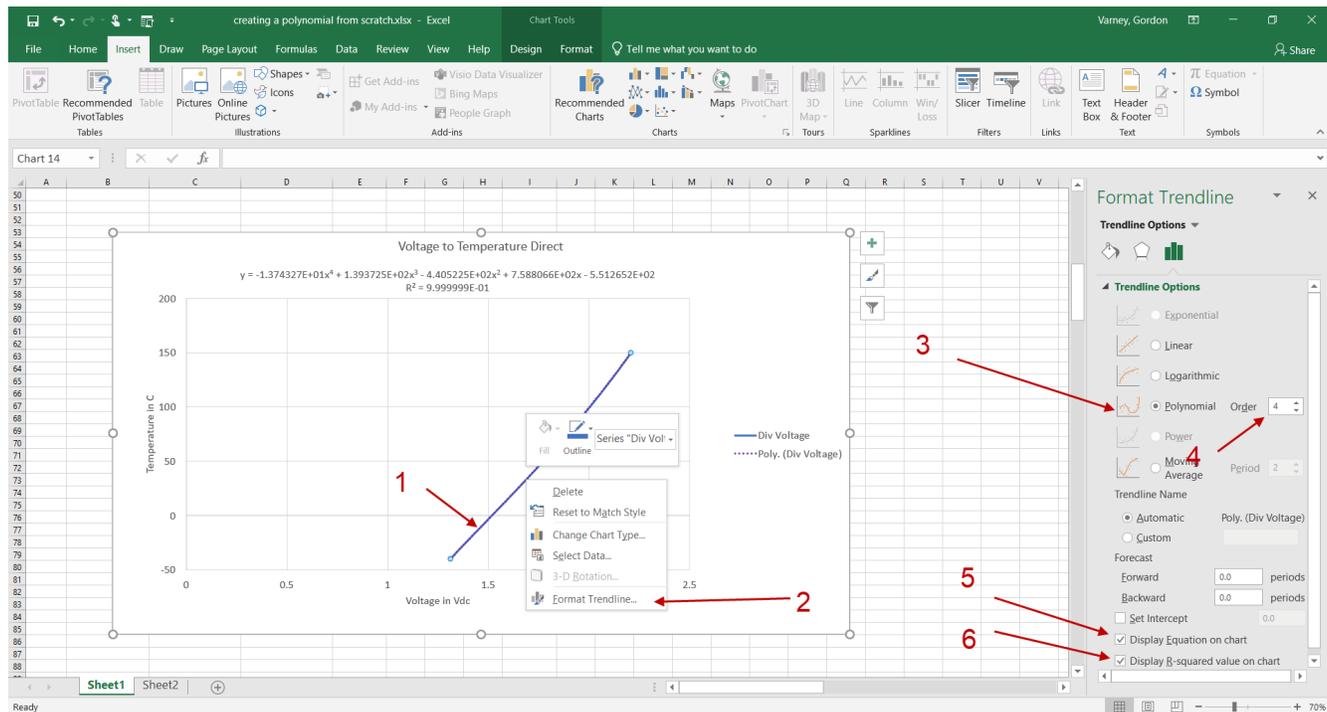


Figure 4-1. Polynomial Flow

Use 6 digits of precision in the coefficients to maintain the accuracy of the calculations as much as possible. [Figure 4-2](#) shows how to change the format of the label to get the required digits for the coefficients.

1. Right click the label for the polynomial and select *Format Trendline Label*
2. Change the category to scientific
3. Change the number of decimal places to "6"

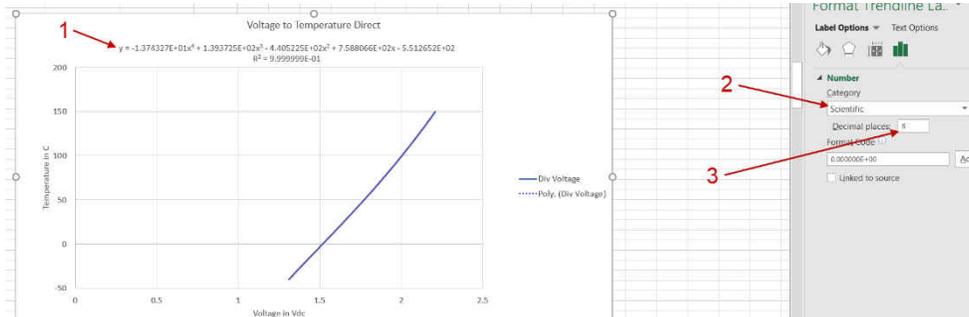


Figure 4-2. Format the Trendline Label

Now you can copy the formula and the coefficients from the trendline label and paste it as text in a spreadsheet.

$$Y = -1.374327E + 01x^4 + 1.393725E + 02x^3 - 4.405225E + 02x^2 + 7.588066E + 02x - 5.512652E + 02 \quad (1)$$

$$T^{\circ}C = A4 \times (V^4) + A3 \times (V^3) + A2 \times (V^2) + A1 \times (V) + A0 \quad (2)$$

where

- V = ADC Voltage
- A4 = -1.374327E+01
- A3 = 1.393725E+02
- A2 = -4.405225E+02
- A1 = 7.588066E+02
- A0 = -5.512652E+02

Remember that if the formula requires the subtraction of a number, keep the (-) minus symbol with the coefficient. Now the polynomial formula can be created in code with only addition as in the main polynomial in [Equation 1](#). Adding a negative number is the same as subtracting.

Now replace the X4 through X values with the measured ADC voltage in the polynomial formula. Use the coefficients you created in the new formula.

5 C Code

Figure 5-1 illustrates the polynomial equation in C code.

EXAMPLE OF C CODE Copy the content of the box below, then paste into your C code

```

// 4th order polynomial equations to calculate the temperature of the thermistor
// C code examples only (NOTE: this code example is based on floating point math)

float VBias = 3.30 ; // set the VBIAS voltage
unsigned int ADC_BITS = 16384 ; // set the number of bits based on you ADC (214 of ADC Bit Value)
float VTEMP = 0; // set up the variable for the measured voltage
float THRM_TEMP = 0; // setup the variable for the calculated temperature

float Thermistor(int raw_ADC) // send the ADC bit value to the calculation function
{
    // THRM calculations - 4th order polynomial regression
    VTEMP = 0; // reset these variables to zero in order to recalculate the new factors
    THRM_ADC = raw_ADC

    float THRM_A0 = -5.475344E+02 ;
    float THRM_A1 = 7.507511E+02 ;
    float THRM_A2 = -4.340839E+02 ;
    float THRM_A3 = 1.371151E+02 ;
    float THRM_A4 = -1.345040E+01 ;

    VTEMP = (VBias/ADC_BITS) * THRM_ADC; // calculate volts per bit then multiply that times the ADV value
    THRM_TEMP = (THRM_A4 * pow(f(VTEMP,4)) + (THRM_A3 * pow(f(VTEMP,3))) + (THRM_A2 * pow(f(VTEMP,2))) + (THRM_A1 * VTEMP) + THRM_A0; // 4th order regression to get temperature
    return THRM_TEMP;
}
    
```

Figure 5-1. C Code

6 Creating a Polynomial for an NTC

A polynomial curve fit does not work for an NTC. The main reason is that there are two points of inflection in the curve. A polynomial curve fit can only work with a curve that has only one point of inflection. As Figure 6-1 shows, when an NTC is plotted, the curve has two points of inflection so the fit is not good. Notice that even negative resistances are present in the curve fit.

$$y = 1.927826E-03x^4 - 4.901709E-01x^3 + 4.146868E+01x^2 - 1.438325E+03x + 2.430037E+04$$

$$R^2 = 9.932779E-01$$

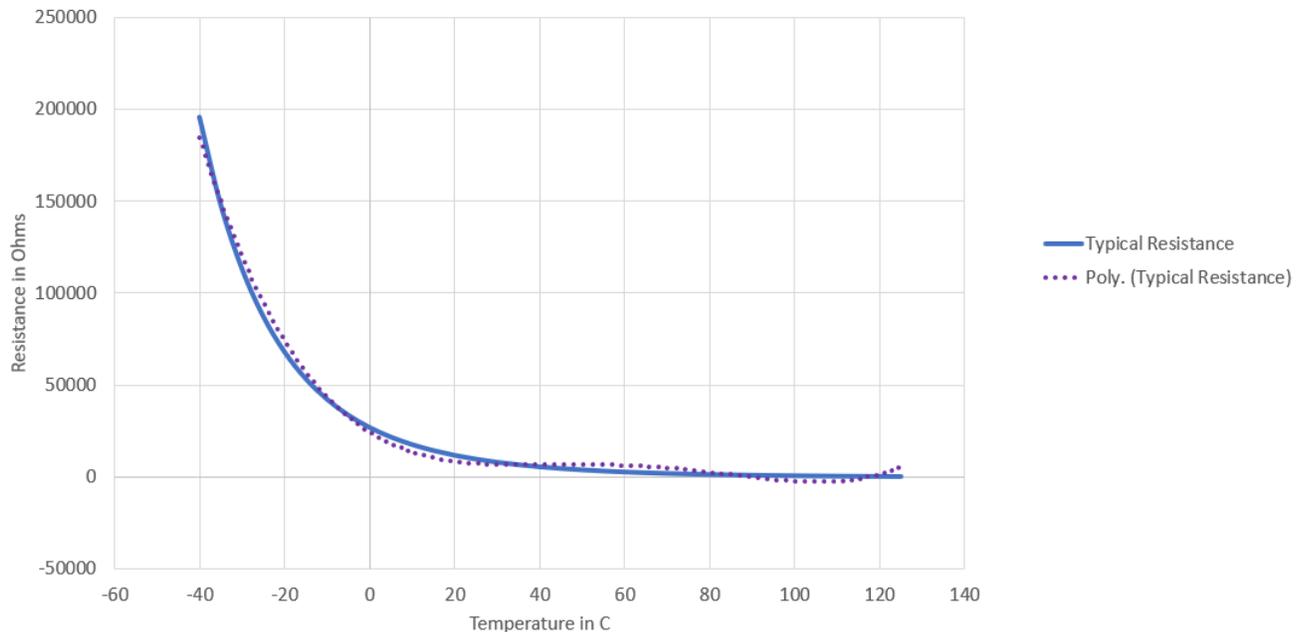


Figure 6-1. NTC Polynomial Curve Fit

7 Summary

A Polynomial curve fit accuracy is seen in the R squared value. The TMP6 has a 0.999999% accuracy. Whereas the NTC has a 0.993% accuracy and negative resistances. This means that the polynomial can provide a temperature from the ADC voltage with increased accuracy and faster processing time for the TMP6 than the Steinhart-Hart equation or when using a look-up table with interpolation. An NTC will not be able to use a polynomial to calculate its temperature.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2022, Texas Instruments Incorporated