

Free licenses available

Note: We currently offer free licenses to people writing project reviews. [Get in touch with us](#) to get your free license before this marketing campaign ends.

1 Abstract

The project offers support for NTC thermistor calculations. The Steinhart-Hart equation is a mathematical model for these thermistors that seems to fit for a wide range of temperatures with high precision. Software to calculate the characteristic Steinhart-Hart coefficients based on temperature-resistance tables for given thermistors as well as functions allowing conversion of temperature values to resistance and vice versa is provided.

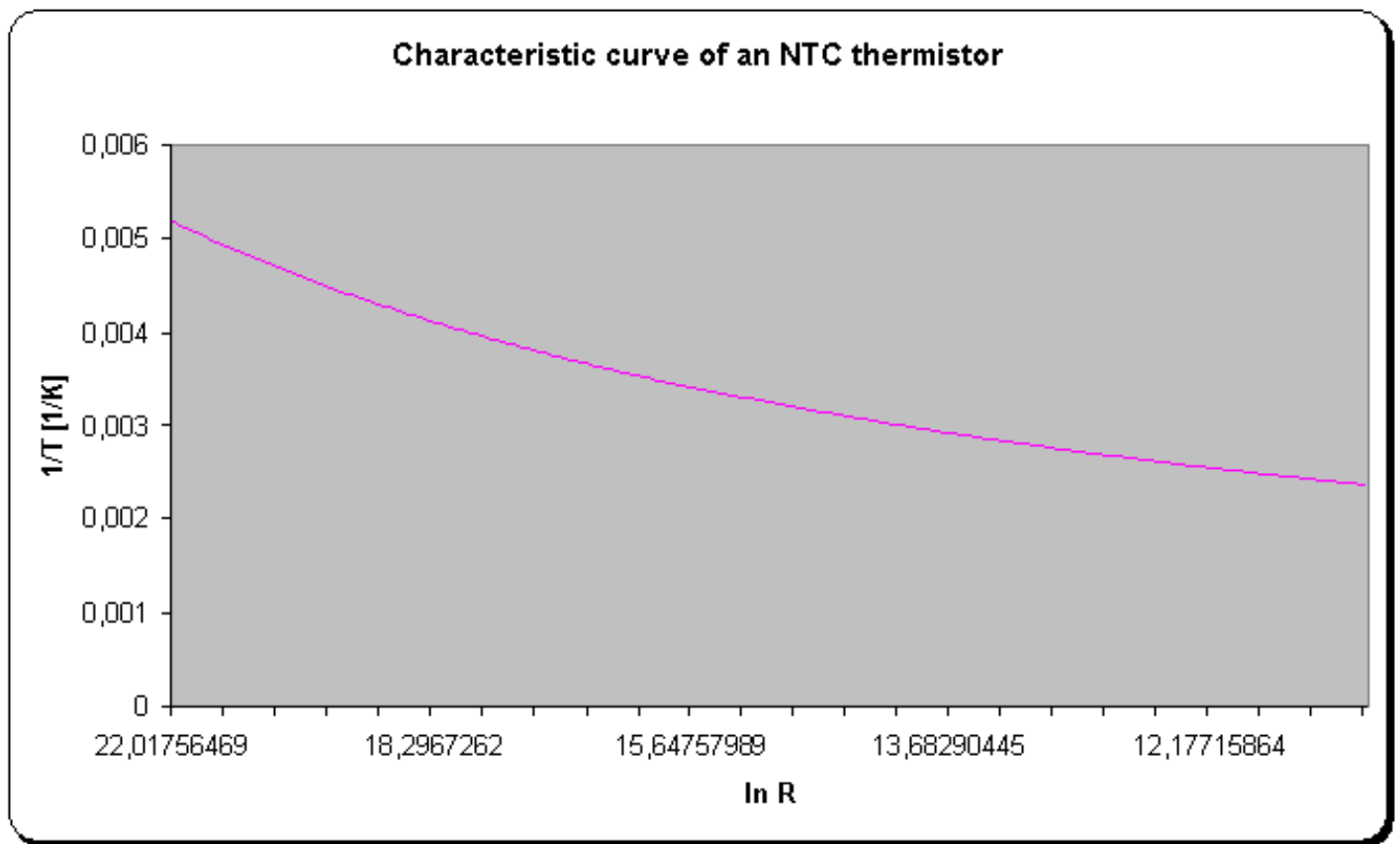
2 Description

A model for the resistivity of a semiconductor as a function of the temperature was found by Steinhart and Hart 1968 ([1]). The Steinhart-Hart law describes the absolute temperature T (in Kelvins) as a function of the NTC thermistor's resistivity (in Ω) according to the formula

$$1/T = a_0 + a_1 \cdot \ln r + a_3 \cdot (\ln r)^3$$

Steinhart-Hart polynom

The figure below shows the typical graph of an NTC thermistors characteristic, giving the reciprocal temperature (in $1 / K$) over the natural logarithm of the resistance (in Ω).



The constants a_0 , a_1 and a_3 , also called Steinhart-Hart coefficients, vary depending on the type of thermistor. To support developer when creating temperature measurement applications, thermistor manufacturer often supply these constants for their products. They also publicate tables where resistivity of thermistor products for a wider range of temperature values are listed.

This project provides software to

- calculate temperature value for a given resistance of an NTC thermistor with given Steinhart-Hart coefficients,
- calculate resistance value for a given temperature for an NTC thermistor with given Steinhart-Hart coefficients and
- evaluate Steinhart-Hart coefficients for an NTC thermistor descibed by a temperature-resistance table.

Apart from the standard Steinhart-Hart equation other forms have been found. For application with lower CPU power a simplified form of the Steinhart-Hart equation can be used.

$$\frac{1}{T} = a_0 + a_1 \cdot \ln r$$

Simplified Steinhart-Hart polynom

On the other hand a quadratic term can be inserted into the formula to increase accuracy giving the extended Steinhart-Hart equation

$$\frac{1}{T} = a_0 + a_1 \cdot \ln r + a_2 \cdot (\ln r)^2 + a_3 \cdot (\ln r)^3$$

Extended Steinhart-Hart polynom

An introduction to thermistors and the Steinhart-Hart polynom can be found at Wikipedia [2].

3 Software

Software is provided by this project for the calculations given in section algorithms. The classes / modules can be used for

- conversion from resistance to temperature,
- conversion from temperature to resistance,
- calculation of Steinhart-Hart coefficients for an NTC thermistor whose characteristics is given as T-R table.

The latter can be done for standard, simplified or extended Steinhart-Hart polynom.

Download the archives for C or Java [here](#).

3.1 Java-Classes

For Java the following classes are available

Class	Description
NtcException	Exception used in this thermistor framework
NtcTable	Represents an T-R table for an NTC thermistor. Provides methods to get name and description of the NTC thermistor as well as methods to add entries and to lookup the resistance for a given temperature value.
NtcThermistorModel	Represents an NTC thermistor modelled using the Steinhart-Hart polynom $1/T = a_0 + a_1 \cdot \ln r + a_3 \cdot (\ln r)^3$. The model will be created using a NtcTable object. It provider methods to read the themistor name, get the NtcTable object, get the Steinhart-Hart coefficients as well as methods for conversion from temperature to resistance and vice versa.
NtcThermistorExtendedModel	Represents an NTC thermistor modelled using the extended Steinhart-Hart polynom $1/T = a_0 + a_1 \cdot \ln r + a_2 \cdot (\ln r)^2 + a_3 \cdot (\ln r)^3$. Derived from NtcThermistorModel thus providing the same methods.
NtcThermistorSimplifiedModel	Represents an NTC thermistor modelled using the simplified Steinhart-Hart polynom $1/T = a_0 + a_1 \cdot \ln r$. Derived from NtcThermistorModel, provides the same methods.

Note: Generics are used in the project so Java 5 or higher is required according to that.

A small test class NtcTest is part of the package together with an example text file containing an T-R table of a theoretical NTC (simu.txt). To make developers life easy the whole application is deployed as an (executable) jar file containing all sources, classes and a manifest file with main class NtcTest.

After download the test application can be invoked by

The output produced can be seen [here](#).

3.2 C Modules

Three C modules are available

Module	Description
coeff.c	Program used to calculate Steinhart-Hart coefficients from an NTC table (given as text file).
ttor.c	Temperature to resistance calculation.
rtot.c	Resistance to temperature calculation.

For usage under Windows they have been compiled (using Cygwin's C compiler). For other OS please compile them with your system C-compiler. An example text file containing an T-R table of a theoretical NTC (simu.txt) is also provided for testing purposes.

Example outout of the "coeff" program is given [here](#).

4 Algorithms

4.1 Conversion from resistance to temperature

Based on a measured resistance value of an NTC thermistor the extended Steinhart-Hart equations allows a simple calculation of the temperature

$$1/T = a_0 + a_1 \cdot \ln r + a_2 \cdot (\ln r)^2 + a_3 \cdot (\ln r)^3$$

Here r is the resistance in Ω and T the absolute temperature in K (K = Kelvin). With absolute zero $T_{abs} = -273.15 \text{ } ^\circ\text{C}$, the formula for the temperature t in $^\circ\text{C}$ finally leads to

$$t = 273,15 \text{ } ^\circ\text{C} + [a_0 + a_1 \cdot \ln r + a_2 \cdot (\ln r)^2 + a_3 \cdot (\ln r)^3]^{-1}$$

By setting appropriate coefficients to zero, the calculation according to the simplified or standard Steinhart-Hart equation can be done as well.

4.2 Conversion from temperature to resistance

To calculate resistance values in Ω from temperature in $^\circ\text{C}$ the root r of the extended Steinhart-Hart equation must be found (for the standard polynom simply set $a_2=0$ in the formulas below)

$$1/T = a_0 + a_1 \cdot \ln r + a_2 \cdot (\ln r)^2 + a_3 \cdot (\ln r)^3$$

with $y = \ln r$, we get

$$1/T = a_0 + a_1 \cdot y + a_2 \cdot y^2 + a_3 \cdot y^3$$

We introduce the substitutions

$$T = t + T_{abs}$$

$$b = a_2 / a_3$$

$$c = a_1 / a_3$$

$$d = (a_0 - 1/T) / a_3$$

$$p = c - \frac{1}{3} \cdot b^2$$

$$q = \frac{2}{27} \cdot b^3 - \frac{1}{3} \cdot b \cdot c + d$$

$$u = \left[-q/2 + (q^2/4 + p^3/27)^{1/2} \right]^{1/3}$$

$$v = \left[-q/2 - (q^2/4 + p^3/27)^{1/2} \right]^{1/3}$$

getting

$$r = e^{u+v-b/3}$$

for the resistance r (in Ω) of a given temperature t (in $^{\circ}\text{C}$).

4.3 Calculation of Steinhart-Hart coefficients

Sometimes calculation of coefficients is done using special temperature values. Inserting four value pairs in the range of interest into the extended Steinhart-Hart polynomial leads to a system of linear algebraic equations (Three value pairs for the standard Steinhart-Hart polynomial). Temperatures typically used are for example 0°C , 15°C , 25°C and 70°C . By solving this system the values for a_0 , a_1 , a_2 and a_3 can be determined.

A better approach is the mathematical optimization technique called ordinary least squares (OLS), that was introduced by Carl Friederich Gauß in 1801. Details about the theory of OLS are given at Wikipedia [\[3\]](#) or MathPlanet [\[4\]](#).

If the function that is approximated is a polynomial, theory of vector spaces, scalar products and orthonormal bases eases the calculation. Given a list with n temperature-resistance pairs

$$(r_0, t_0), (r_1, t_1), \dots, (r_{n-1}, t_{n-1})$$

(where n should be at least 3), the question for best fitting of the Steinhart-Hart coefficients leads to the minimization of

$$\text{sum} := \sum_{i=0}^{n-1} [t(r_i) - t_i]^2$$

where t_i is the i -th temperature value and $t(r_i)$ is the calculated temperature according to the polynomial.

This optimization requires a small mathematical excursion. For a positive integer n and given abscissa values x_0, x_1, \dots, x_{n-1} the polynomials with degree $\leq n$ build a vector space V . On V a scalar product can be defined by

$$[p, q] := \sum_{i=0}^n p(x_i) \cdot q(x_i)$$

By

$$|p| := [p, p]^{1/2}$$

the scalar products makes V a normed space. For a function f with ordinate values

$$y_i := f(x_i)$$

for $i = 0, \dots, n-1$ there is exactly one polynomial p_f in V that matches the function in the coordinates $(x_0, y_0), (x_1, y_1), \dots, (x_{n-1}, y_{n-1})$.

If a positive integer m , with $m \leq n$, is given, all polynomials of degree $\leq m$ build a sub space U of V . The question for the best approximation of f by polynomials from U is equivalent to the best approximation of p_f by polynomial from U in respect to $|\cdot|$.

$$s = \sum_{i=0}^{n-1} [u_f(x_i) - y_i]^2 = \sum_{i=0}^{n-1} [u_f(x_i) - p_f(x_i)]^2 = |u_f - p_f|^2$$

Answering the last question turns out to be very easy, if an orthonormal base $\{u_1, u_2, \dots, u_m\}$ of U is given. For p_f the best approximation u_f evaluates as

$$u_f = \sum_{i=1}^m [u_i, p_f] \cdot u_i$$

If for example an R-T table with value pairs from 0°C up to 100°C in steps of 1°C is given, the polynomial p_f has degree 101 (!). Fortunately it must not be found, in order to find the Steinhart-Hart coefficients and thus finding u_f . Starting with the canonical base $\{v_1=1, v_2=x, v_3=x^2, v_4=x^3\}$ of U , an orthonormal base $\{u_1, u_2, u_3, u_4\}$ can be evaluated. After that, the coefficients of u_f are calculated by the sum stated above.

The coordinates

$$(x_0, y_0), (x_1, y_1), \dots, (x_{n-1}, y_{n-1})$$

are calculated from the temperature-resistance pairs as

$$x_i = \ln r_i$$

$$y_i = 1/(t_i - T_{abs})$$

for $i = 0, \dots, n-1$.

The orthonormal base of U is evaluated recursively. We set

$$u_1 = v_1 = 1$$

and

$$w_i = v_i - \sum_{j=1}^{i-1} [u_j, u_i] \cdot u_j$$

$$u_i = w_i / |w_i|^{1/2}$$

for $i = 2, \dots, m = 4$.

With this base we can calculate u_f as

$$u_f = \sum_{i=1}^m [u_i, p_f] \cdot u_i$$

and determine the Steinhart-Hart coefficients from this equation. (Note that the scalar product can be calculated without knowing p_f as stated above).

5 License

The software of this project is published under the GNU Lesser General Public License.

6 References

[1] J.S. Steinhart and S.R. Hart, "Calibration Curves for Thermistors," Deep Sea Research, Vol. 15, 497-503, 1968.

[\[2\] Thermistors, NTCs and Steinhart-Hart equation on Wikipedia](#)

[\[3\] Least squares on Wikipedia](#)

[\[4\] Least square on PlanetMath](#)