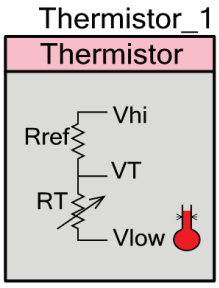
热敏电阻计算器

1.0

特性



- 适用于大多数的负温度系数 (NTC) 热敏电阻
- 具有查找表 (LUT) 或公式实现方法
- 可选参考电阻，基于热敏电阻值
- 可选温度范围
- LUT 方法具有可选计算分辨率

概述

热敏电阻计算器组件是基于测量从热敏电阻提供的电压，来计算温度。此组件适用于大多数 NTC 热敏电阻。它基于温度范围和相应的用户提供的参考电阻来计算 Steinhart-Hart 公式的系数。此组件提供使用生成的系数以基于测量的电压值返回温度值的 API 函数。

此组件不使用内部的模数转换器或 AMUX，因此需要在工程中单独安置这个组件。

何时使用热敏电阻计算器

此组件只有一个用例。组件提供的 API 用于基于从热敏电阻测量的电压值来计算温度。

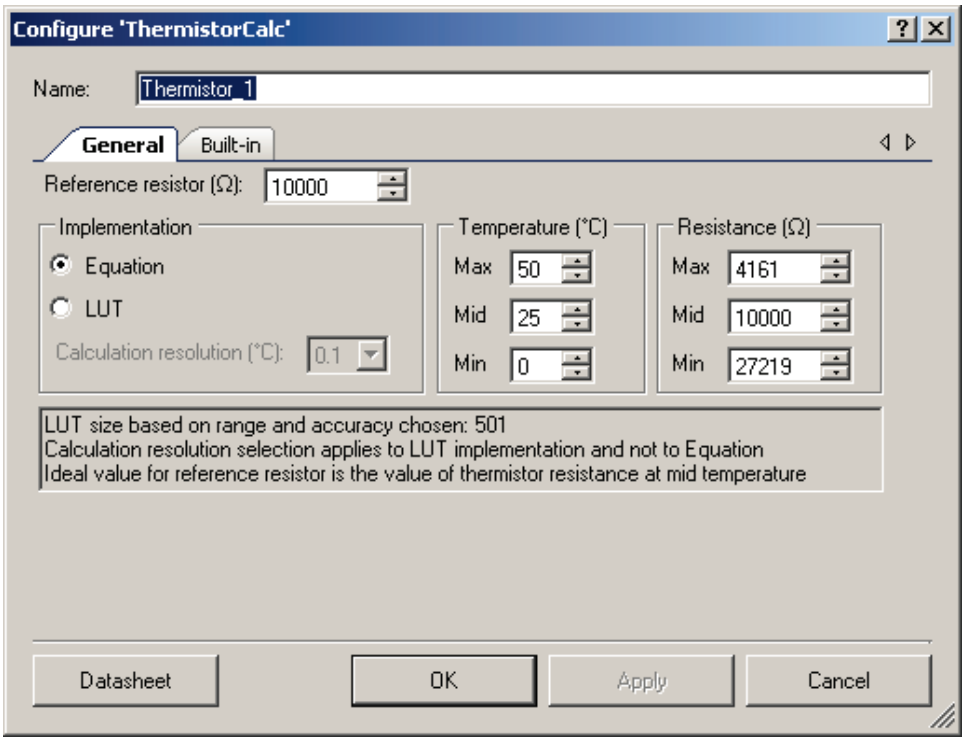
输入/输出连接

这是一个软件组件，没有任何输入/输出连接。

参数和设置

将热敏电阻计算器组件拖入设计中，双击它以打开 Configure（配置）对话框。该对话框有一个选项卡，可引导您完成热敏电阻计算器器件的设置。

一般选项卡



General（一般）选项卡提供以下参数。

Reference resistor（参考电阻）

Reference resistor（参考电阻）连接至热敏电阻，如符号所示，用于进行恒压类型的温度测量。Rref 和 RT 可以相互交换以随着温度的升高而增大或减小电压值。理想情况下，参考电阻的值应等于要求的温度范围处于中间温度时的热敏电阻值。

范围 = 1Ω 到 1GΩ（默认值 10000）。

实现

可以通过 Equation（公式）或 LUT 方法 获得温度。这两种方法的权衡因素是存储器、速度、范围和分辨率。Equation（公式）方法更加准确，有固定的范围和分辨率。Equation（公式）方法使用较多的内存，因为它需要浮点数学库。LUT 使用较少的内存，而且响应时间更快。默认为 Equation（公式）。

Calculation resolution (°C)（计算精度 (°C)）

如果选择 LUT 方法，此参数将生效以提供 LUT 温度精度。

此参数指定的精度是温度测量的精度。这意味着此精度对应于电压到温度的转换。它不考虑系统中的其他误差，例如参考电阻的容差、参考电压的变化或模数转换器的精度。

假设一个准确的电压测量，此参数提供此器件的温度输出的精度。

选项 = 0.01、0.05、0.1（默认值）、0.5、1、2 °C

Temperature (°C)（温度 (°C) ） / Resistance (Ω)（电阻 (Ω)）

第一列用于指定需要的温度范围的最大温度、中间温度和最小温度。第二列用于输入与相应的温度关联的电阻。Steinhart-Hart 系数是基于此表格中的条目计算得出的。

这些参数还确定了 LUT 实现的温度范围。这些参数中输入的最高和最低温度值构成了 LUT 的开始值和结束值。

范围：

- 温度（最大、中间、最小）-80 到 325 °C。（默认值：最大 = 50，中间 = 25，最小 = 0）
- 电阻（最大、中间、最小）0 到 1MΩ（默认值：最大 = 4161，中间 = 10000，最小 = 27219）

尽管此组件支持宽泛的温度范围（-80 到 325 °C），建议使用标准化范围（-40 到 125 °C）以获得更高精度的结果。

有关标准化温度范围，通常可以在使用的特定热敏电阻的基本介绍中找到定义温度的精密电阻值。有关非标准化温度范围（超出 -40 到 125 °C 范围），您需要执行预测量以获得特定温度的准确电阻值。Steinhart-Hart 系数在标准化范围内提供最高的精度。标准化范围以外的任何其他范围将提供较低的精度。

如果您使用热敏电阻计算器器件在宽泛的范围（甚至于标准化范围）内测量温度，强烈建议将宽泛的范围拆分为较小的子范围以获得最高精度。例如，范围 -40 到 125 °C 应拆分为三个子范围：-40° 到 0°、0° 到 50°、50° 到 125°。拆分此范围使每个子范围中的电阻/温度曲线更具线性，从而消除 LUT 条目中的误差，同时提供了最高的温度精度。如果您有多个子范围，则您需要确定使用哪个实例来计算电压。电阻计算与温度范围无关，因此可以根据任一实例使用函数计算电阻。那么，基于计算的电阻，适用子范围的实例必须用于计算温度。

信息

其他详细信息将基于其他提供参数，例如 LUT 的大小，以及要选择的合适的参考电阻。LUT 的大小显示在第一行中，并仅限于 2001 个步幅。LUT 的大小由以下公式确定：

LUT _ SIZE = (MAX _TEMP − MIN _TEMP) / 计算精度 + 1

如公式所示，LUT 的大小取决于范围和精度。因此，当 LUT 的大小超出了限制，精度旁边将显示错误，说明需要降低精度或范围。

应用程序编程接口

应用程序编程接口 (API)函数允许您使用软件配置组件。下表列出了每个函数的接口，并进行了说明。以下各节将更详细地介绍每个函数。

默认情况下，PSoC Creator 将实例名称“Thermistor_1”分配给提供的设计中的第一个器件实例。您可以将其重命名为遵循标识符语法规则的任何唯一值。实例名称会成为每个全局函数名称、变量和常量符号的前缀。出于可读性考虑，下表中使用的实例名称为“Thermistor”（热敏电阻）。

函数	说明
uint32 Thermistor_GetResistance (int16 Vreference, int16 VThermistor)	参考电阻和热敏电阻上的电压的数字值被作为参数传递给此函数。它们可视为此器件的输入。此函数将基于电压值返回（输出）电阻。
int16 Thermistor_GetTemperature (uint32 ResT)	热敏电阻的值被作为参数传送至此函数。此函数将基于电阻值返回（输出）温度。用于计算温度的方法取决于是选择了 Equation（公式）还是 LUT。

uint32 Thermistor_GetResistance(int16 Vreference, int16 VThermistor)

说明： 参考电阻和热敏电阻上的电压的数字值被作为参数传递给此函数。它们可视为此器件的输入。此函数将基于电压值返回（输出）电阻。

参数： Vreference 是参考电阻上的电压。

Vthermistor 是热敏电阻上的电压。此函数使用这两个电压的比例。因此，这两个参数的单位必须相同。

返回值： 返回值是热敏电阻中的电阻。返回类型是 32 位无符号整数，如上述提供的函数原型所示。返回的值是电阻，单位为欧姆。

副作用： None

int16 Thermistor_GetTemperature (uint32 ResT)

说明： 热敏电阻的值被作为参数传送至此函数。此函数将基于电阻值返回（输出）温度。用于计算温度的方法取决于是选择了 Equation（公式）还是 LUT。

参数： ResT 是热敏电阻中的电阻，单位为欧姆。

返回值： 返回值是温度，单位为 1/100ths 摄氏度。例如，当实际温度为 23.45 摄氏度时，返回值为 2345。

副作用： None

固件源代码示例

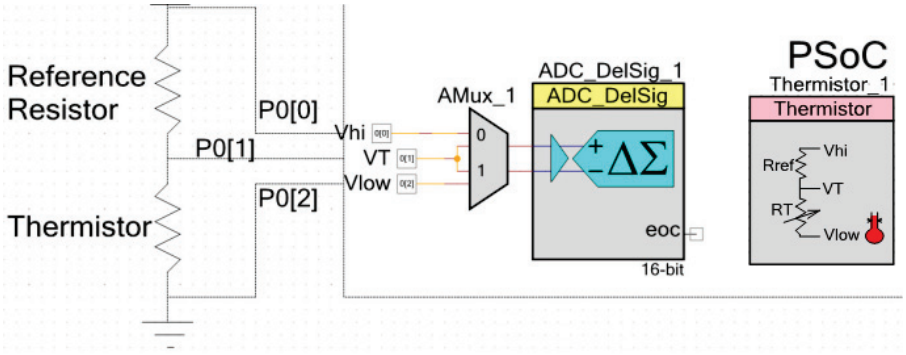
PSoC Creator 提供了大量包括原理图和示例代码的示例项目。

有关更多信息，请参考 AN66477 - PSoC 3 和 PSoC 5 使用热敏电阻测量温度。

功能描述

整个工程需要外部电阻和热敏电阻连接至 PSoC。外部连接的电阻和热敏电阻的电压将使用模数转换器测得，这些值将通过 API 调用被传送至热敏电阻器件。此 API 调用的返回值为温度。将模数转换器从此器件中排除的方法使模数转换器可供项目中的其他函数使用。整个系统的框图，请参阅下图。

图 1. 基于温度监控系统的热敏电阻的框图



恒压 Vhi 施加于电阻和热敏电阻的组合之上，如图 1 所示。

热敏电阻的电阻随着温度的变化而变化，因此热敏电阻上的电压降随着温度而变化。测得热敏电阻和电阻上的电压降，使用以下公式找到热敏电阻的电阻

$$R_T = R_{ref}\left(\frac{V_T - V_{low}}{V_{hi} - V_T}\right)$$

然后，基于此电阻通过 Equation（公式）或 LUT 方法确定温度。在 Equation（公式）方法中，温度通过直接使用如下所示的 Steinhart-Hart Equation（公式）获得：

$$\frac{1}{T_K} = A + B * ln(R_{Thermistor}) + C * (ln(R_{Thermistor}))^3$$

其中，A、B、C 是由此器件计算的 Steinhart-Hart 系数。这些系数是通过求解替代此器件中的“热敏电阻参数”而构成的三个联立方程而获得的。

如果使用的是 LUT 方法，此器件将生成与温度和电阻相关的表格，且此表格将存储在程序存储器中。要生成 LUT，使用以下公式计算范围内的各种温度的电阻。

$$R_{Thermistor} = e^{\left[(\beta-(\alpha/2))^{\frac{1}{3}} - (\beta+(\alpha/2))^{\frac{1}{3}}\right]}$$

用户指明，从“Min Temp”（最低温度）、和“Accuracy”（精度）的增量一直至“Max Temp”（最高温度），计算相应的电阻。LUT 由此器件使用这些电阻构成，且 LUT 将存储在程序存储器中。然后，在运行期间，从此表格中获得测量的特定电阻对应的温度。

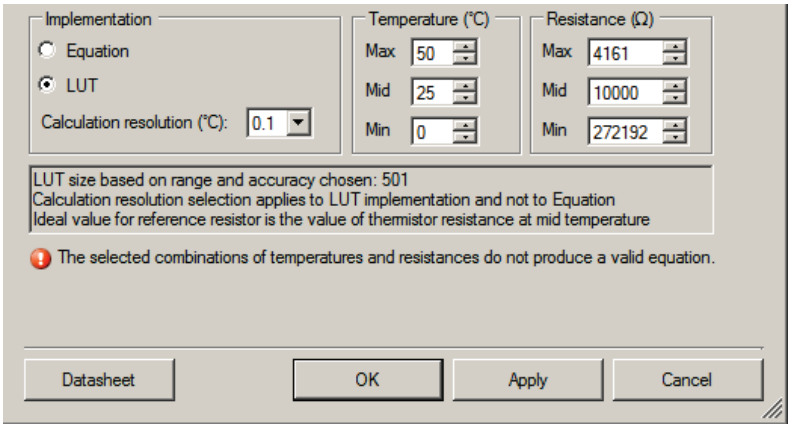
其中：$$\alpha = \frac{A - \frac{1}{T}}{C}, \beta = \sqrt{\left(\frac{B}{3C}\right)^3 + \frac{\alpha^2}{4}}$$

这些公式在本文中有提供，供用户参考之用。此组件执行所有这些计算，并基于 Configure（配置）对话框中的选择提供需要的温度。

这两种实现方法的比较如下表所示。

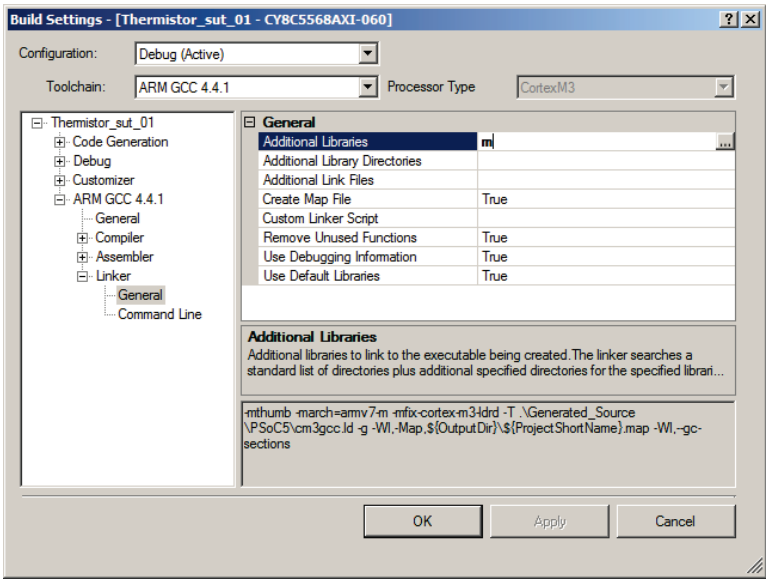
	公式	LUT	注释
精度 (+/-)	0.01	≥ 0.01	显示的精度仅为计算的精度。它不包括热敏电阻、参考电阻、电压参考或模数转换器的精度。公式方法的精度高于 ± 0.01 °C，但是输出仅限于精度 ± 0.01 °C，因为函数返回 1/100ths °C。
内存使用情况	如果未包括浮点，则较高	如果未包括浮点，则较低	Equation（公式）的内存使用情况是固定的，它归因于浮点库。如果浮点库已经被其他器件或函数所使用，则 Equation（公式）方法是高效的。 LUT 的内存使用情况取决于选择的范围和精度。
	如果已包括浮点，则较低	如果已包括浮点，则较高	
范围	比指定范围宽	仅限于指定范围	在 Equation（公式）中，可以在指定范围以外测量温度（精度降低）。LUT 仅限于指定范围。
速度	更慢。	更快。	Equation（公式）方法使用计算密集型的浮点数学库函数。LUT 方法仅需要 LUT 的二进制搜索。

并非所有的温度与电阻的组合都生成有效的 Steinhart-Hart 等式。如果输入的值生成无效的公式，将生成以下错误：



使用热敏电阻基本介绍或使用预先准确测量的值时，不应出现此情况。

在借助 GCC 编译器使用公式实现方法时，您必须在 Build Settings（构建设置）对话框的 **Linker**（连接器）选项中通过在 **Additional libraries**（其他库）字段中输入“m”来指定包括数学库，如下图所示。



资源

在固件中完全实现此组件。它不会消耗任何其他 PSoC 资源。

API 内存使用情况

组件使用情况显著不同，取决于编译器、设备、使用 API 的数量以及组件配置。下表提供了给定的组件配置中的所有 API 的内存使用情况。

已使用 Release（发布）模式中配置的关联编译器进行了测量，此编译器使用了 Size（大小）的最佳设置。有关特定的设计，可分析编译器生成的映射文件以确定内存使用情况。

配置	PSoC 3 (Keil_PK51)		PSoC 5 (GCC)		PSoC 5LP (GCC)	
	Flash（闪存）字节	SRAM 字节	Flash（闪存）字节	SRAM 字节	Flash（闪存）字节	SRAM 字节
公式	339	0	208	0	208	0
LUT	774 + (LUT 大小 * 4)	0	144 + (LUT 大小 * 4)	0	144 + (LUT 大小 * 4)	0

性能

此组件的性能取决于自定义程序中选择的实现方法。已使用 Release （发布）模式中配置的关联编译器使用 24 MHz 的 CPU 速度收集以下测量。这些数字应视为近似值，并应用于确定必要的权衡。

器件	Equation（公式）方法	101 个条目的 LUT	2001 个条目的 LUT
PSoC 3 (Keil PK51)	26,000 个周期	3400 个周期	5400 个周期
PSoC 5 (GCC)	20,000 个周期	200 个周期	540 个周期

组件更改

版本	更改说明	更改原因及影响
1.0	版本 1.0 是热敏电阻计算器组件的首次发行版	

© 赛普拉斯半导体公司，2012。此处所包含的信息可能会随时更改，恕不另行通知。除赛普拉斯产品的内嵌电路之外，赛普拉斯半导体公司不对任何其他电路的使用承担任何责任。也不根据专利或其他权利以明示或暗示的方式授予任何许可。除非与赛普拉斯签订明确的书面协议，否则赛普拉斯产品不保证能够用于或适用于医疗、生命支持、救生、关键控制或安全应用领域。此外，对于可能发生运转异常和故障并对用户造成严重伤害的生命支持系统，赛普拉斯不授权将其产品用作此类系统的关键器件。若将赛普拉斯产品用于生命支持系统中，则表示制造商将承担因此类使用而招致的所有风险，并确保赛普拉斯免于因此而受到任何指控。

PSoC® 是赛普拉斯半导体公司的注册商标，PSoC® Creator™ 和 Programmable System-on-Chip™ 是赛普拉斯半导体公司的商标。此处引用的所有其他商标或注册商标归其各自所有者所有。

所有源代码（软件和/或固件）均归赛普拉斯半导体公司（赛普拉斯）所有，并受全球专利法规（美国和美国以外的专利法规）、美国版权法以及国际条约规定的保护和约束。赛普拉斯据此向获许可者授予适用于个人的、非独占性、不可转让的许可，用以复制、使用、修改、创建赛普拉斯源代码的派生作品、编译赛普拉斯源代码和派生作品，并且其目的只能是创建自定义软件和/或固件，以支持获许可者仅将其获得的产品依照适用协议规定的方式与赛普拉斯集成电路配合使用。除上述指定的用途之外，未经赛普拉斯的明确书面许可，不得对此类源代码进行任何复制、修改、转换、编译或演示。

免责声明：赛普拉斯不针对此材料提供任何类型的明示或暗示保证，包括（但不仅限于）针对特定用途的适销性和适用性的暗示保证。赛普拉斯保留在不做出通知的情况下对此处所述材料进行更改的权利。赛普拉斯不对此处所述之任何产品或电路的应用或使用承担任何责任。对于可能发生运转异常和故障并对用户造成严重伤害的生命支持系统，赛普拉斯不授权将其产品用作此类系统的关键器件。若将赛普拉斯产品用于生命支持系统中，则表示制造商将承担因此类使用而招致的所有风险，并确保赛普拉斯免于因此而受到任何指控。

产品使用可能受适用的赛普拉斯软件许可协议限制。