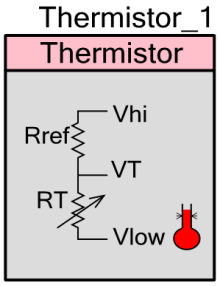
热敏电阻计算器

1.20

特性



- 适应于大多数的负温度系数（NTC）热敏电阻
- 支持查找表 （LUT） 或公式计算方法
- 可选的参考电阻值
- 可选温度范围
- LUT 实现方法支持可选计算精度

概述

热敏电阻计算器组件通过测量热敏电阻两端的电压值来计算温度。该器件适用于大多数 NTC 热敏电阻。它利用用户设定的温度范围和参考电阻阻值来计算 Steinhart-Hart 方程系数。该组件提供 API 函数，该函数根据计算得到的 Steinhart-Hart 方程系数返回基于测量电压值的温度值。

此器件不使用内部的模数转换器或 AMUX，因此需要在您的项目中单独放置这些器件。

何时使用热敏电阻计算器

此器件只有一个使用案例。组件提供的 API 通过热敏电阻两端的电压测量值来计算温度。

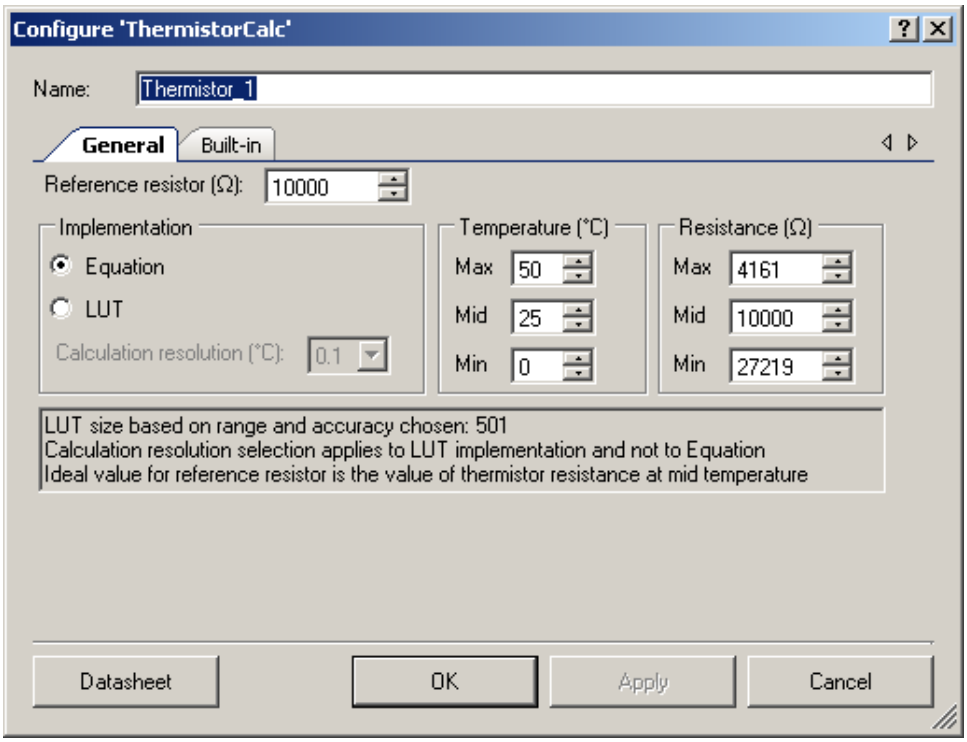
输入/输出连接

这是一个纯软件组件，没有任何输入/输出接口。

参数和设置

将热敏电阻计算器组件拖入设计中，双以打开 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组件的设置。

General（常规）选项卡



General 选项卡提供以下参数。

Reference resistor（参考电阻）

Reference resistor 连接至热敏电阻，如符号所示，用于进行测量恒压类型的温度。Rref 和 RT 可以相互交换位置，以获得一个随温度增大或减小的电压值。理想情况下，参考电阻值应等于所设温度范围中间值对应的热敏电阻的阻值。

电阻范围为 1 Ω 至 1 GΩ（默认值 10000）。

Implementation（实现）

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

Calculation resolution（°C）（计算精度）

如果您选择了 LUT 实现，此参数可设置 LUT 方法的温度计算精度。

此参数指定的精度是温度测量的精度，这意味着该精度与电压到温度的转换对应。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。

假设那是一个准确的电压测量，那么此参数提供组件的温度输出精度。

选项 = 0.01、0.05、0.1（默认值）、0.5、1、2 °C

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

第一列用于指定所需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有关的电阻。Steinhart-Hart 方程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 的起始值和结束值。

范围：

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

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

有关标准化温度范围，您通常可以在您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（-40 °C 以上到 125 °C），您需要执行预测量以获得特定温度的准确电阻值。Steinhart-Hart 方程系数提供在标准化范围内的最高精度。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组件在宽泛的范围（甚至于标准化范围）内测量温度，强烈建议将宽泛的范围分为较小的子范围以获得最高精度。例如，-40 到 125 °C 的范围应分为三个子范围：

-40 °C 到 0 °C、0 °C 到 50 °C 和 50 °C 到 125 °C. 拆分该范围使每个子范围中的电阻/温度曲线更加平滑，从而消除 LUT 条目中的误差，同时提供最高的温度精度。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，因此可以使用任意一个实例中的函数来计算电阻。那么，根据计算得出的电阻，必须使用适用于子范围的实例来计算温度。

信息

根据其他参数，如 LUT 的大小和合适的参考电阻，来提供其他详细信息。LUT 的大小显示在第一行中，并仅限于 2001 步。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 的大小超出了范围，精度旁边将显示错误，说明需要降低精度或范围。

应用编程接口（API）

通过应用编程接口（API）子程序，您可以使用软件对该组件进行配置。下表列出并说明了每个函数的接口。以下各节将更加详细地介绍每个函数。

默认情况下，PSoC Creator 将实例名称“Thermistor_1”分配给指定设计中组件的第一个实例。您可以将其重新命名为遵循标识符语法规则的任何唯一值。该实例名称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。它们可视为该组件的输入。该函数将根据电压值返回（输出）电阻。
int16 Thermistor_GetTemperature (uint32 resT)	热敏电阻的值被作为参数传送至该函数。该函数将根据电阻值返回（输出）温度。用于计算温度的方法取决于选择Equation（公式）还是LUT方法。

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。它们可视为该组件的输入。该函数将根据电压值返回（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。该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。

其他影响: 无

int16 Thermistor_GetTemperature (uint32 resT)

说明： 热敏电阻的值被作为参数传送至该函数。该函数将根据电阻值返回（输出）温度。用于计算温度的方法取决于选择Equation（公式）还是LUT方法。

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1/100摄氏度。例如，当实际温度为23.45摄氏度时，返回值为2345。

其他影响: 无

示例固件源代码

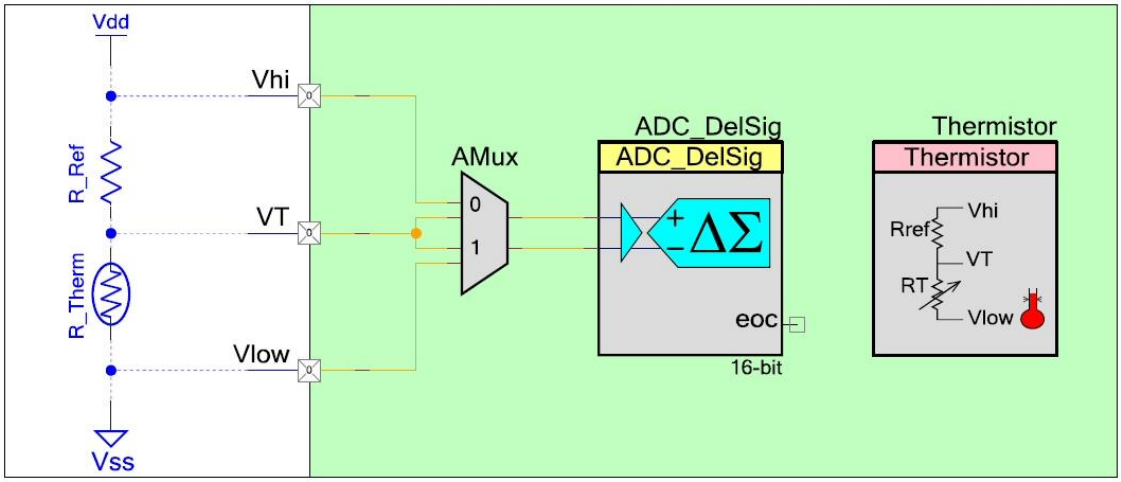
在“Find Example Project”（查找示例项目）对话框中，PSoC Creator 提供了大量的示例项目，包括原理图和示例代码。要获取组件特定的示例，请打开组件目录中的对话框或原理图中的组件实例。要查看通用示例，请打开“Start Page”（起始页）或“**File**”（文件）菜单中的对话框。根据要求，可以通过使用对话框中的“**Filter Options**”（滤波器选项）来限定可选的项目列表。

有关更多信息，请参考 PSoC Creator 帮助中的“Find Example Project”（查找示例项目）主题。

功能说明

整个项目需要外部参考电阻和热敏电阻连接至 PSoC。外部连接的参考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。将模数转换器从该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 适用于参考电阻和热敏电阻的组合，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将随着温度变化而变化。测得热敏电阻和参考电阻上的电压下降，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，根据该电阻，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 方程获得：

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

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

如果使用 LUT 方法，此组件将生成与温度和电阻相关的表格，并将其存储在程序存储器中。要想生成 LUT，使用以下公式计算范围内的各种温度的电阻。

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

根据用户指定的“Min Temp”（最低温度）、“Accuracy”（精度）的增量以及“Max Temp”（最高温度），计算出相应的电阻。LUT 由该组件使用这些电阻构成并存储在程序存储器中。

然后，在运行期间，从此表格中获得与测量得到的特定电阻相应的温度。

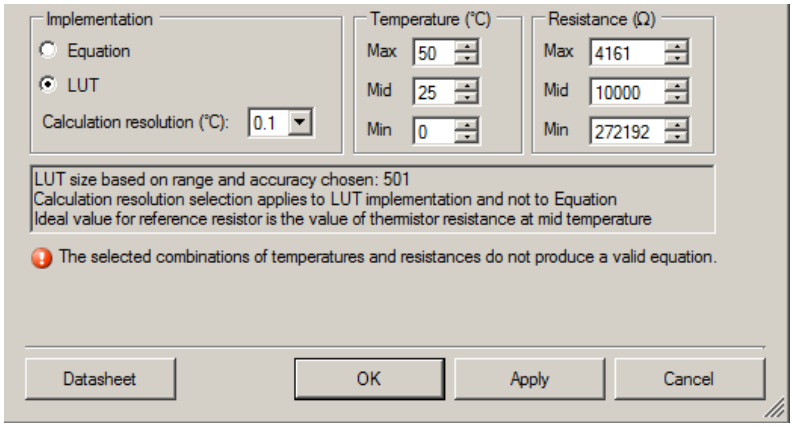
其中：$$\alpha = \frac{A - \frac{1}{T}}{C}, \beta = \sqrt{\left(\frac{B}{3C}\right)^3 + \frac{\alpha^2}{4}}$$

这些公式在本文中提供，供用户参考之用。该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所需的温度。

这两种实现方法的比较如下表所示。

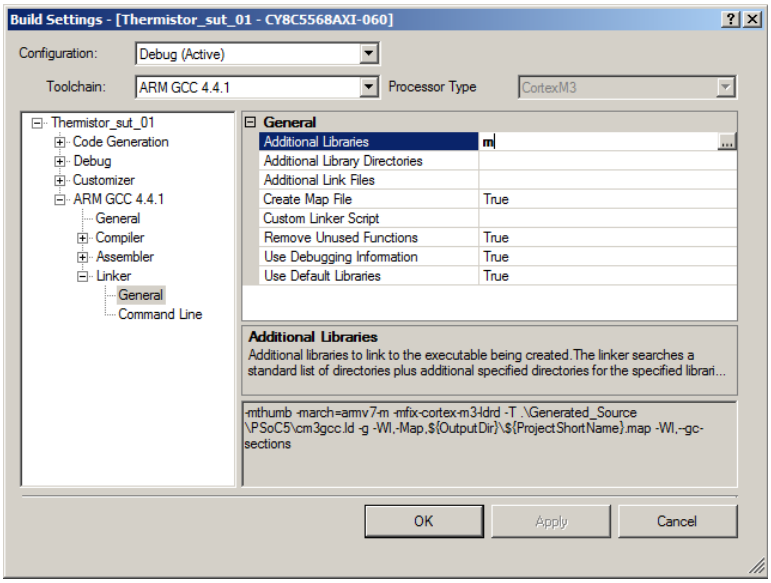
	公式	LUT	注释
精度（+/-）	0.01	≥ 0.01	显示的精度仅为计算的精度。它不包括热敏电阻、参考电阻、电压参考或模数转换器的精度。Equation（公式）方法的精度高于± 0.01 °C，但是输出仅限于精度± 0.01 °C，因为函数返回1/100 °C。
储存器的使用情况	如果未包括浮点，则较高	如果未包括浮点，则较低	Equation（公式）的储存器使用情况是固定的，它归因于浮点库。如果浮点库已经被其他组件或函数所使用，则 Equation（公式）方法是高效的。LUT的储存器使用情况取决于选择的范围和精度。
	如果已包括浮点，则较低	如果已包括浮点，则较高	
范围	比指定范围宽	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。

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



当使用热敏电阻数据手册中介绍的参考值或使用预先测量的准确值时，不会出现这种情况。

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



MISRA 合规性

本节介绍了MISRA-C:2004合规性和本组件的偏差情况。定义了下面两种类型的偏差：

- 项目偏差 — 适用于所有 PSoC Creator 组件的偏差
- 特定偏差 — 仅适用于该组件的偏差

本节提供了有关组件特定偏差的信息。*系统参考指南*的“MISRA 合规性”章节中介绍了项目偏差以及有关 MISRA 合规性验证环境的信息。

热敏电阻计算器组件没有任何特定偏差。

使用资源

此组件完全由固件实现。它不会消耗任何其他 PSoC 资源。

API 的储存器使用情况

根据编译器、器件、所使用的 API 数量以及组件的配置不同，组件对存储资源的占用也不一样。下表提供了在某种器件配置中所有 API 占用存储器的大小。

数据是在将编译器设置为 Release 模式并将优化等级设置为 Size 的情况下测得的。对于特定的设计，分析完编译器生成的映射文件后可以确定组件占用存储器的大小。

配置	PSoC 3 （Keil_PK51）		PSoC 4 （GCC）		PSoC 5LP（GCC）	
	闪存 字节	SRAM 字节	闪存 字节	SRAM 字节	闪存 字节	SRAM 字节
公式	329	0	252	0	236	0
LUT	764 + (LUT 大小 * 4)	0	168 + (LUT 大小 * 4)	0	188 + (LUT 大小 * 4)	0

性能

该器件的性能取决于定制器中选择的实现方法。已使用 Release 模式中配置的关联编译器使用 24 MHz 的 CPU 速度收集以下测量值。这些数字应视为近似值，并用于确定必要的权衡。

器件	Equation（公式）方法	101个元素的LUT	2001个元素的LUT
PSoC 3 （Keil PK51）	27,000个周期	4000个周期	6000个周期
PSoC 4 （GCC）	30,000个周期	280个周期	650个周期
PSoC 5LP （GCC）	20,000个周期	250个周期	600个周期

器件的更改

版本	更新内容	更改原因/影响
1.20.b	Minor datasheet edit.	
1.20.a	更新了数据手册。	移除了已停产的PSoC 5器件的参考内容。
1.20	删除了Thermistor_GetResistance()函数中校验的参数。	防止热敏电阻断开时CPU停止工作。
	更新了“MISRA合规性”章节。	该组件没有任何特定偏差。
	更新了“API 存储器使用情况”章节。	
	更新了数据手册中的存储器使用情况和PSoC 4组件的性能。	
1.10	更新了“样品固件源代码”章节，描述如何查找PSoC Creator示例项目	
	更新了基于温度监控系统的热敏电阻的框图。	旧图的质量较差。
	更新了API存储器使用情况和性能数据。	
	将Thermistor_GetResistance()的API参数名称从Vreference、VThermistor更改为vReference、vThermistor；Thermistor_GetTemperature()的API参数名称从ResT更改为resT。	根据赛普拉斯编码标准，参数应为骆驼式小写形式。
	添加了“MISRA合规性”章节。	该组件未通过MISRA合规性验证。
1.0	版本 1.0 是热敏电阻计算器组件的首次发行版	