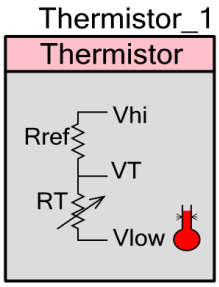
サーミスタ計算機
1.0

特長



- NTC サーミスタ(負温度係数サーミスタ)のほとんどに適用可能
- ルックアップテーブル(LUT)または方程式による実装
- サーミスタの値に基づき基準抵抗を選択
- 選択可能な温度範囲
- LUT 法では計算分解能を選択可能

概要

サーミスタ計算機コンポーネントは、サーミスタで測定された電圧に基づき温度を計算します。本コンポーネントはほとんどの NTC サーミスタに適用可能です。温度範囲と対応する基準抵抗に基づき、Steinhart-Hart 式係数を計算します。本コンポーネントは生成された係数を利用し測定された電圧に基づく温度を返す API 関数を提供します。

本コンポーネントは ADC または AMUX を含みません。これらのコンポーネントはプロジェクトに別途配置しなければなりません。

サーミスタ計算機の用途

このコンポーネントの用途は一つです コンポーネントの API により、サーミスタで測定された電圧に基づき温度を計算します。

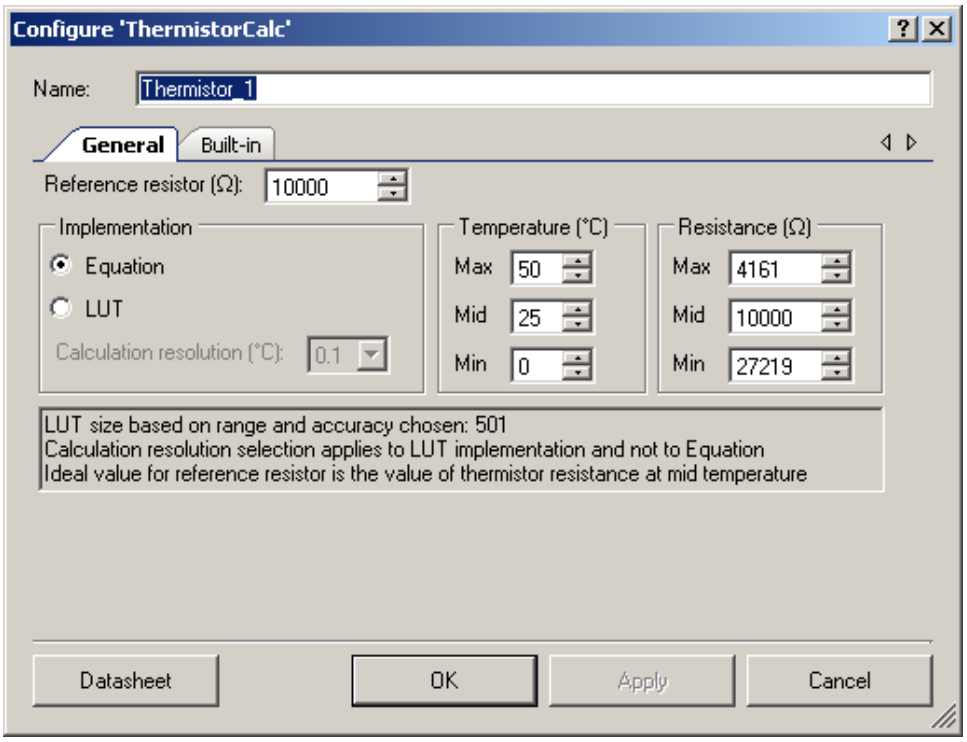
入出力の接続

このコンポーネントはソフトウェアコンポーネントであり入出力の接続はありません。

パラメータおよび設定

サーミスタ計算機コンポーネントを回路図の上にドラッグし、ダブルクリックして Configure ダイアログを開きます。このダイアログにはサーミスタ計算機コンポーネントを設定するタブがあります

General タブ



General タブには、次のパラメータがあります。

Reference resistor

Reference resistor は、シンボルに表示されている通りサーミスタに接続して、温度測定の電圧タイプを一定にします。Rref と RT を交換すると、温度に対する電圧変化を反転させることができます。理想的には基準抵抗の値を、所望の温度範囲の中間でサーミスタの値と等しくなるようにします。

範囲 = 1Ω～1GΩ (初期値 10000)。

Implementation

方程式または LUT により温度を取得します。2 つの方法のトレードオフは、メモリ、速度、測定範囲、分解能です。方程式法はより正確で、測定範囲および精度が固定されています。方程式法では浮動小数点演算ライブラリが必要であるため、より多くのメモリを使用します。LUT はメモリをそれほど使用せず、応答も高速です。初期設定は方程式法です。

Calculation resolution (°C)

LUT 実装を選択するとこのパラメータが有効になり、温度精度の設定ができます。

このパラメータで指定する精度は、温度測定の精度です。この精度は、電圧から温度への変換そのものに対するものです。つまり基準抵抗の許容誤差、基準電圧の変動、ADC の精度といったシステム内の他の不正確さを含みません。

電圧測定が正確であると想定した場合、このパラメータは本コンポーネントの温度出力の精度となります。

選択肢 = 0.01、0.05、0.1(初期値)、0.5、1、2°C

Temperature (°C) / Resistance (Ω)

最初の列は、所望の温度範囲の最高、中間、最低温度を指定します。次の列は、その温度に対する抵抗値を入力します。この表への入力に基づき Steinhart-Hart 係数が計算されます。

またこれらのパラメータは、LUT 実装の温度範囲も決定します。これらのパラメータに入力された最高および最低温度値により、LUT の開始値および終了値が決まります。

範囲:

- 温度(max, mid, min) -80～325°C。(初期値: max = 50、mid = 25、min = 0)
- 抵抗値(max, mid, min) 0～1MΩ (初期値 max = 4161、mid = 10000、min = 27219)

本コンポーネントは広範な温度範囲(-80～325°C)をサポートしていますが、より良い精度を得るためには標準温度範囲(-40～125°C)を使用することを推奨します。

標準温度範囲内では、精密な抵抗値がサーミスタのデータシートの抵抗値温度特性表に記載されています。-40～125°C の外側の非標準温度範囲については、事前測定を行なって、特定の温度について正しい抵抗値を取得しなければなりません。Steinhart-Hart 係数は、標準の範囲内で最良の精度を提供します。標準範囲を超える場合は、精度が落ちる場合があります。

サーミスタ計算機コンポーネントを使用してより広い範囲の温度を測定する場合、標準範囲内であっても、その広い温度範囲を小区間に分割して精度を最大限に高めることを強く推奨します。例えば範囲が-40～125°C である場合、以下の 3 つの小区間に分割します: -40°C～0°C、0°C～50°C、50°C～125°C。範囲を分割することで、各小区間の抵抗温度曲線をより直線状にし、LUT 入力の不正確さを平滑化し、最良の温度精度が得られます。小区間がいくつかある場合、どのインスタンスを電圧計算のために利用するかを決定する必要があります。抵抗値の計算は温度範囲から独立していますので、どのインスタンスの関数も抵抗値を計算するのに利用可能です。その後計算された抵抗値に基づき、該当する小区間のインスタンスを利用して温度を計算しなければなりません。

Information

LUT のサイズや適切な基準抵抗といったその他のパラメータに基づき、追加情報が表示されます。LUT サイズは冒頭行に表示されます。サイズは 2001 ステップに限定されています。LUT サイズは以下の方程式により決定されます:

LUT _ サイズ = (最高_温度 − 最低_温度) / 計算分解能 + 1

式に示される通り、LUT サイズは温度範囲および精度によって変ります。したがって、LUT サイズが限度を超える場合、および精度は別にしてエラーが起こる場合は、精度または範囲のいずれかを下げる必要があります。

アプリケーションプログラミングインタフェース

アプリケーションプログラミングインターフェース (API) ルーチンにより、ソフトウェアを使用してコンポーネントを設定できます。次の表は、各関数へのインターフェースとその説明を示しています。続くセクションでは、各関数について詳しく説明します。

初期設定では、PSoC Creator は、ユーザの回路図に最初に配置されたコンポーネントのインスタンス名として"Thermistor_1"を割り当てます。インスタンスの名称は、識別子の文法ルールに従って固有の名前に変更できます。インスタンス名は、すべてのグローバル関数名、変数名、定数名の接頭辞になります。便宜上、次の表では"Thermistor"というインスタンス名を使用します。

関数	機能
uint32 Thermistor_GetResistance (int16 Vreference, int16 VThermistor)	基準抵抗とサーミスタ全体の電圧のデジタル値が、パラメータとしてこの関数に渡されます。これらは本コンポーネントへの入力と見なされます。関数は電圧に基づき抵抗値を返します。
int16 Thermistor_GetTemperature (uint32 ResT)	サーミスタの抵抗値が、パラメータとしてこの関数に渡されます。関数は抵抗値に基づき温度を返します。温度を計算する方法は、方程式法またはLUT法により異なります。

uint32 Thermistor_GetResistance(int16 Vreference, int16 VThermistor)

機能: 基準抵抗およびサーミスタ全体の電圧のデジタル値が、パラメータとしてこの関数に渡されます。これらは本コンポーネントへの入力と見なされます。関数は電圧に基づき抵抗値を返します。

パラメータ: Vreferenceは基準抵抗両端の電圧です。

Vthermistorはサーミスタ両端の電圧です。これらの2つの電圧の比率をこの関数で使用します。したがって、両方のパラメータの単位は同じでなければなりません。

返り値: 返り値はサーミスタの抵抗値です。返り値の型は、上記の関数プロトタイプに示すように、32ビットの符号なし整数です。返り値は抵抗値(単位オーム)です。

注意事項: なし

int16 Thermistor_GetTemperature (uint32 ResT)

機能: サーミスタの抵抗値が、パラメータとしてこの関数に渡されます。関数は抵抗値に基づき温度を返します。温度を計算する方法は、方程式法またはLUT法により異なります。

パラメータ: ResTはサーミスタの抵抗値(単位オーム)です。

返り値: 返り値は温度(摂氏、0.01°C単位)です。例えば、実際の温度が23.45°Cである場合、返り値は2345です。

注意事項: なし

ファームウェアソースコードのサンプル

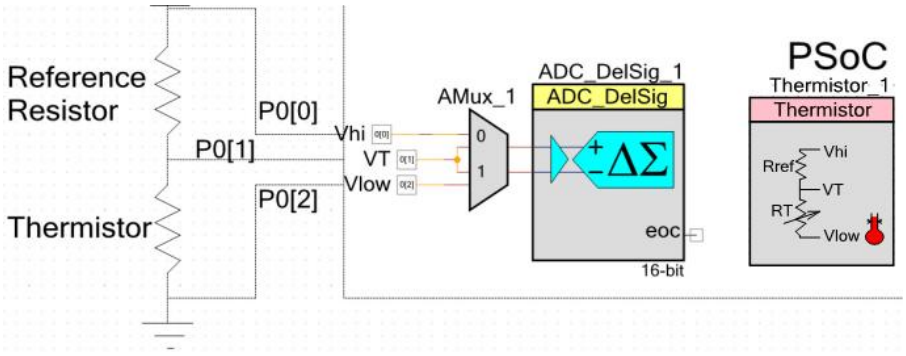
PSoC Creator は、数多くのサンプルプロジェクトを提供しており、そこには回路図およびコード例が含まれています。

詳しくは AN66477 - PSoC 3 and PSoC 5 Temperature Measurement with Thermistor を参照してください。

機能の詳細

プロジェクト全体では、外部に抵抗とサーミスタを PSoC に接続することが必要です。外部に接続する抵抗とサーミスタの電圧は ADC を利用して測定され、その値は API 呼び出しを通してサーミスタコンポーネントへと渡されます。この API 呼び出しの返り値は温度です。この方式は ADC をコンポーネントに含まず拘束しないため、ADC をプロジェクト内の他の関数で利用することができます。システム全体のブロック図を次に示します。

図 1. サーミスタによる温度モニタシステムのブロック図



プロジェクト全体では、外部に抵抗とサーミスタを PSoC に接続することが必要です。外部に接続する抵抗とサーミスタの電圧は ADC を利用して測定され、その値は API 呼び出しを通してサーミスタコンポーネントへと渡されます。この API 呼び出しの返り値は温度です。この方式は ADC をコンポーネントに含まず拘束しないため、ADC をプロジェクト内の他の関数で利用することができます。システム全体のブロック図を次に示します。

図 1 に示す抵抗とサーミスタの組合せに、一定電圧 Vhi が加えられます。

サーミスタの抵抗値は温度変化により変ります。したがってサーミスタの電圧降下は温度により変ります。サーミスタおよび抵抗全体の電圧降下を測定し、以下の式を利用してサーミスタの抵抗値を特定できます。

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

温度は方程式法または LUT 法のいずれかを通して、抵抗値に基づき決定されます。温度は、以下の Steinhart-Hart 方程式を直接使用して、方程式法で得られます:

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

A、B、C は、本コンポーネントにより計算される Steinhart-Hart 係数です。係数は、本コンポーネント内に “Thermistor Parameters"を代入することで生成される 3 つの連立方程式を解くことで得られます。

LUT 法の場合、本コンポーネントにより温度と抵抗値の表が生成され、プログラムメモリに保存されます。LUT を生成するには、当該範囲内の温度に対する抵抗値を以下の式を利用して計算します:

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

最低温度から抵抗値を計算し始めて、ユーザー定義の最高温度まで精度単位で上げていきます。LUT はこれらの抵抗値からなり、プログラムメモリに保存されます。測定された抵抗値に対応する温度は、動作中に表から取得します。

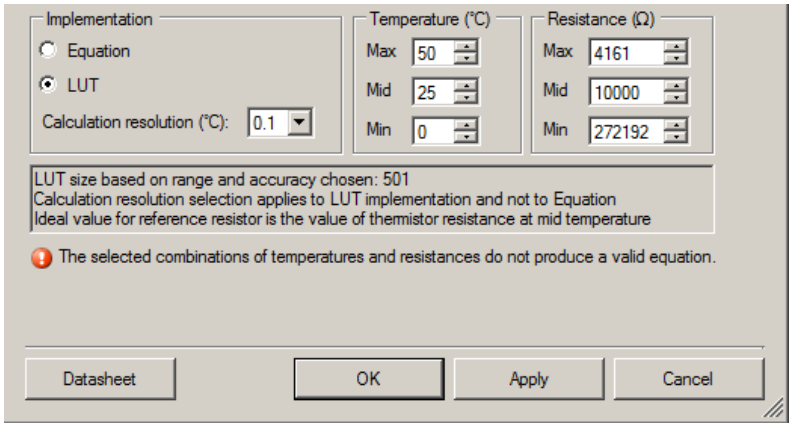
$$\text{ここに } \alpha = \frac{A - \frac{1}{T}}{C}, \beta = \sqrt{\left(\frac{B}{3C}\right)^3 + \frac{\alpha^2}{4}}$$

上記の式はユーザの参考として記載します。本コンポーネントはこれらのすべての計算を行い、Configure ダイアログでの選択に基づき必要な温度を提供します。

以下の表は、2 つの実装方法の比較を示したものです。

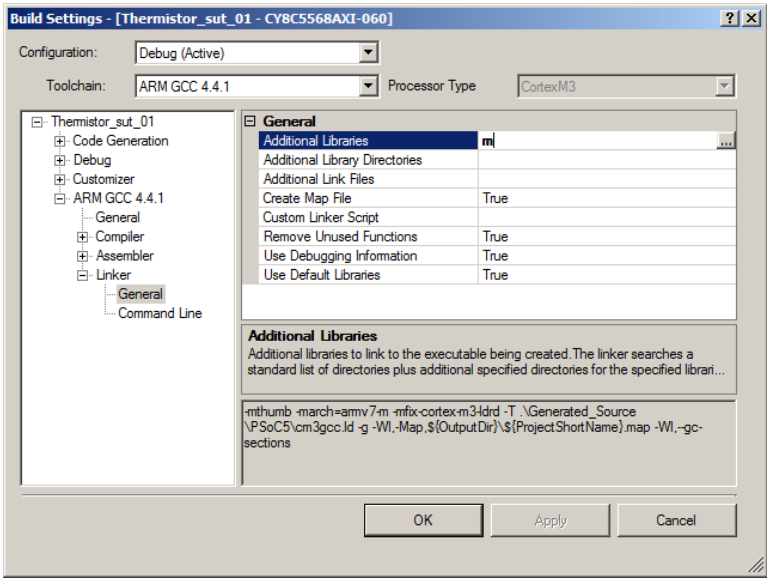
	方程式法	LUT法	コメント
精度(+/-)	0.01	≥ 0.01	精度には、計算のみの精度が示されています。これには、サーミスタ、基準抵抗、基準電圧、ADCの精度は含まれていません。方程式法の精度は±0.01℃以下ですが、関数が摂氏0.01℃単位の数を返すため、出力は±0.01℃の分解能に制限されます。
メモリ使用量	浮動小数点がまだ含まれていない場合、より高くなります。	浮動小数点がまだ含まれていない場合、より低くなります。	方程式法のメモリ使用量は固定されており、それは浮動小数点ライブラリが原因です。別のコンポーネントまたは関数により浮動小数点ライブラリがすでに利用されている場合、方程式法は効率的です。 LUT法のメモリ使用量は、選択する範囲および精度により異なります。
	浮動小数点がすでに含まれている場合、より低くなります。	浮動小数点がすでに含まれている場合、より高くなります。	
範囲	指定範囲より広い	指定範囲に限定	方程式法では、温度を指定範囲の外で測定可能です(精度は下がります)。LUT法は指定範囲に限定されています。
速度	遅い	速い	方程式法は、計算を多用する浮動小数点演算ライブラリを使用します。LUT法は、LUTのバイナリサーチを行うだけです。

温度と抵抗値をどのように組合せても、有効な Steinhart-Hart 方程式となるわけではありません。入力した値が無効な方程式を生成した場合には、以下のエラーが発生します:



これは、サーミスタデータシートの基準値を利用する場合、または正確に事前測定された値を利用する場合には、起こりません。

GCC コンパイラで方程式法を実装する場合、下図に示すように Build Settings ダイアログの **Linker** オプションにある **Additional libraries** フィールドに"m"を入力して、演算ライブラリをインクルードしなければなりません。



リソース

コンポーネントはファームウェアで完全に実装されています。他の PSoC リソースは消費しません。

API メモリ使用量

コンポーネントのメモリ使用量は、コンパイラ、デバイス、使用されている API の数やコンポーネントの構成によって大きく異なります。以下の表は、特定のコンポーネント構成で使用できるすべての API についてのメモリ使用量を示しています。

最適化をサイズ優先、リリースモードに設定したコンパイラを使って測定されました。特定の設計については、コンパイラによって生成されたマップファイルを分析してメモリ使用量を特定できます。

構成	PSoC 3 (Keil_PK51)		PSoC 5 (GCC)		PSoC 5LP (GCC)	
	フラッシュ バイト	SRAM バイト	フラッシュ バイト	SRAM バイト	フラッシュ バイト	SRAM バイト
式	339	0	208	0	208	0
LUT	774 + (LUT サイズ*4)	0	144 + (LUT サイズ*4)	0	144 + (LUT サイズ*4)	0

性能

本コンポーネントの性能は、カスタマイザで選択する実装方法により異なります。以下の値は、CPU 速度 24MHz、コンパイラをリリースモードに設定した状態で測定されました。これらの数値は、必要なトレードオフを特定するための近似値として使用してください。

デバイス	方程式法	101エントリLUT	2001エントリLUT
PSoC 3 (Keil PK51)	26,000サイクル	3,400サイクル	5,400サイクル
PSoC 5 (GCC)	20,000サイクル	200サイクル	540サイクル

変更履歴

バージョン	変更の説明	変更の理由 / 影響
1.0	バージョン1.0はサーミスタ計算機コンポーネントの最初のリリースです。	

Copyright © 2005-2012 Cypress Semiconductor Corporation 本文書に記載される情報は、予告なく変更される場合があります。Cypress Semiconductor Corporation は、サイプレス製品に組み込まれた回路以外のいかなる回路を使用することに対しても一切の責任を負いません。特許又はその他の権限下で、ライセンスを譲渡又は暗示することもありません。サイプレス製品は、サイプレスとの書面による合意に基づくものでない限り、医療、生命維持、救命、重要な管理、又は安全の用途のために使用することを保証するものではなく、また使用することを意図したものでもありません。さらにサイプレスは、誤動作や故障によって使用者に重大な傷害をもたらすことを合理的に予想される、生命維持システムの重要なコンポーネンツとしてサイプレス製品を使用することを許可していません。生命維持システムの用途にサイプレス製品を提供することは、製造者がそのような使用におけるあらゆるリスクを負うことを意味し、その結果サイプレスはあらゆる責任を免除されることを意味します。

PSoC Designer™及び Programmable System-on-Chip™は、Cypress Semiconductor Corp.の商標、PSoC®は同社の登録商標です。本文書で言及するその他全ての商標又は登録商標は各社の所有物です。

全てのソースコード(ソフトウェア及び/又はファームウェア)は Cypress Semiconductor Corporation (以下「サイプレス」)が所有し、全世界(米国及びその他の国)の特許権保護、米国の著作権法並びに国際協定の条項により保護され、かつそれらに従います。サイプレスが本書面によるライセンシーに付与するライセンスは、個人的、非独占的かつ譲渡不能のライセンスであって、適用される契約で指定されたサイプレスの集積回路と併用されるライセンシーの製品のみをサポートするカスタムソフトウェア及び/又はカスタムファームウェアを作成する目的に限って、サイプレスのソースコードの派生著作物を複製、使用、変更、そして作成するためのライセンス、並びにサイプレスのソースコード及び派生著作物をコンパイルするためのライセンスです。上記で指定された場合を除き、サイプレスの書面による明示的な許可なくして本ソースコードを複製、変更、変換、コンパイル、又は表示することは全て禁止されます。

免責条項:サイプレスは、明示的又は黙示的を問わず、本資料に関するいかなる種類の保証も行いません。これには、商品性又は特定目的への適合性の黙示的な保証が含まれますが、これに限定されません。サイプレスは、本文書に記載される資料に対して今後予告なく変更を加える権利を留保します。サイプレスは、本文書に記載されるいかなる製品又は回路を適用又は使用したことによって生ずるいかなる責任も負いません。サイプレスは、誤動作や故障によって使用者に重大な傷害をもたらすことが合理的に予想される生命維持システムの重要なコンポーネンツとしてサイプレス製品を使用することを許可していません。生命維持システムの用途にサイプレス製品を提供することは、製造者がそのような使用におけるあらゆるリスクを負うことを意味し、その結果サイプレスはあらゆる責任を免除されることを意味します。

ソフトウェアの使用は、適用されるサイプレスソフトウェアライセンス契約によって制限され、かつ制約される場合があります。