

PSoC 4 を使用した AFE の実装

著者 : Dineshbabu Mani

関連製品ファミリ : PSoC® 4100PS

関連するサンプルコード : CE223621、CE223620、CE223618

さらに多くのサンプルコードが必要な場合は、以下を参照してください。

増え続ける数百もの PSoC のサンプルコードについては、[サンプルコードの Web ページ](#)にアクセスしてください。ここで PSoC 4 ビデオライブラリを探索することもできます。

AN223616 は、PSoC® 4 を使用して、センシングアプリケーションにアナログフロントエンド (AFE) を実装する方法を説明しています。電圧、抵抗、および容量の出力用の 3 タイプのセンサでの測定に一般的に使用される AFE の実装について説明します。

目次

1	はじめに.....	1	5	静電容量センサ用 AFE.....	10
2	さまざまなタイプのセンサ用 AFE (アナログフロント エンド).....	2	5.1	単一勾配法.....	10
3	電圧出力センサ用 AFE.....	2	5.2	シグマデルタ変調方式.....	11
3.1	実装.....	3	5.3	精度分析.....	14
4	抵抗センサ用 AFE.....	4	6	まとめ.....	14
4.1	電流源メソッド.....	4	7	関連資料.....	14
4.2	レシオメトリック抵抗分割器法.....	5		改訂履歴.....	16
4.3	精度分析.....	7		セールス、ソリューションおよび法律情報.....	17

1 はじめに

PSoC 4 は、プログラマブル AFE を信号処理エンジンに統合するスケラブルで再構成可能なアーキテクチャを提供することにより、センサベースのシステムの設計を簡素化します。PSoC 4 は、センサデータを取集し、前処理し、フォーマットした後にシリアル通信インターフェースを介して、ホストプロセッサに送信する設計を可能にします。

アナログ センサは一般的に出力電気信号に応じて、電圧、電流、抵抗、静電容量、またはインダクタンスの 5 つの異なるタイプがあります。各センサはそのタイプに応じた AFE 設計を必要とします。以下に例を示します。

- 熱電対 (電圧出力の温度センサ) は計測用アンプを必要とします。
- 環境光センサ (電流出力センサ) はトランスインピーダンス アンプ (TIA) を必要とします。

複数のアナログセンサを使用するシステムは、通常、AFE を実装するために複数の専用 IC を必要とし、これにより BOM コストと PCB サイズが増加します。IoT アプリケーション用に設計されたシステムは、複数のセンサからのデータを組み合わせて、一般にセンサフュージョンと呼ばれる新しいセンシング機能に対応する必要があります。多くの場合、センサフュージョンソリューションにはカスタム AFE が必要です。PSoC 4 は特殊な IC の必要性を減らし、シングルチップソリューションでカスタム AFE を作成する機能を提供します。

[図 1-4](#)に、センサベースのシステムの一般的なブロックダイアグラムを示します。それは以下のものを含みます。

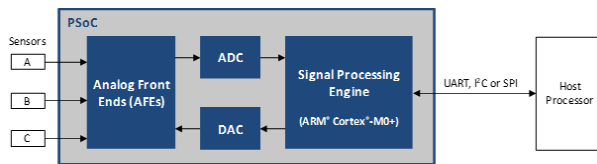
Commented [SS(I1)]: Described on above overview. So no need to add.

Formatted: Font: (Default) MS Gothic, (Asian) MS Gothic, Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

1. 信号を増幅およびフィルタリングしてセンサ出力を調整する AFE。
2. 調整されたセンサ出力をデジタルデータに変換するアナログデジタルコンバータ (ADC) またはコンパレータ (図示せず)。
3. センサデータをフォーマットしてホストプロセッサに送信するための、シリアル通信インターフェースを備えたプログラム可能な信号処理エンジン。信号処理エンジン (32 ビット Arm[®]Cortex[®]-M0+) は、ソフトウェアで AFE を校正および調整できます。

図 1. センサベースのシステム



このアプリケーションノートでは、電圧、抵抗、および容量出力センサのさまざまな AFE 実装方法について説明します。また、各タイプのセンサのパフォーマンス分析についても説明します。最後に、サンプルコードのリファレンスを使用して、これらのセンサを PSoc 4 とインターフェースする方法を説明します。

関連資料セクションには、学習を促進するための豊富なドキュメント類が用意されています。PSoc 4100PS を初めて使用する場合は、AN79953、PSoc 4 入門を参照して、デバイスを理解し、シンプルなデザインを作成する方法を学習してください。

2 さまざまなタイプのセンサ用 AFE (アナログフロントエンド)

アナログセンサは、出力パラメータに応じて、抵抗性、誘導性、容量性、電圧、または電流の 5 つのタイプに分類されます。これらのセンサの出力は、その物理特性に基づいて変化します。たとえば、サーミスタでは、抵抗は温度に基づいて変化します。各センサタイプには適切な AFE が必要です。

出力には 5 つのタイプがありますが、マイクロコントローラ (MCU) で測定できる基本的な電気パラメータは、電圧、時間、および周波数です。センサの出力は、これらの 3 種類の電氣的パラメータのいずれかに変換され、相当するデジタルカウント値として測定されます。たとえば、サーミスタでは、温度によってセンサの抵抗が変化します。サーミスタを基準抵抗器と直列に配置して、定電圧源によって駆動される抵抗分割器を形成できます。次に、サーミスタと基準抵抗器間の電圧降下を測定できます。サーミスタの抵抗は、測定された電圧降下値と既知の基準抵抗から計算できます。

このアプリケーションノートで説明する各 AFE は、特定の方法を使用して、センサ出力を基本的な電氣的パラメータ (電圧、時間、または周波数) に変換します。

3 電圧出力センサ用 AFE

電圧出力センサでは、センサ端子間の電圧は物理量に応じて変化します。電圧出力センサの例は次のとおりです。

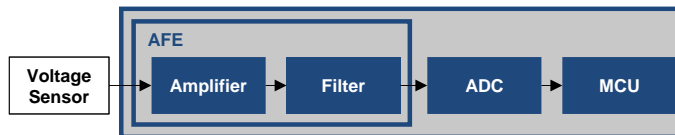
- 熱電対は、温度によって電圧出力が変化します。
- ホール効果センサは、隣接または内蔵の導体に流れる電流から発生する磁場によって電圧出力が変化します。
- PIR モーションセンサは、赤外線放射物体がセンサの前を通過すると、電圧出力が変化します。

図 2 図 2 に、電圧出力センサの信号チェーンの一般的なブロックダイアグラムを示します。

Formatted: Font: (Default) MS Gothic, (Asian) MS Gothic, Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

図 2. 電圧測定 AFE ブロックダイアグラム



センサの出力は最初にアンプに接続されます。アンプの機能は、ADC の入力範囲に合わせて入力信号を増幅することです。たとえば、センサの最大電圧出力が 30 mV で、ADC の範囲が 1 V である場合、アンプのゲインを 32 とすると信号を 0.96 V に増幅するために、オフセット電圧、信号オーバーシュートなどに対して 0.04 V のマージンが許されます。既に ADC の測定範囲内にある信号の場合、アンプはユニティゲインバッファにするか、アンプ自体が不要な場合もあります。

多くのセンサには、高入力インピーダンスのアンプが必要です。センサ信号が充分高くアンプ段をバイパスできる場合、センサの出力は、高い入力インピーダンスを提供するユニティゲインバッファを介して ADC に接続できます。熱電対や電流シャントなどのセンサには、差動アンプが推奨されます。出力インピーダンスが低く、出力電圧信号が大幅に高い差動センサの場合、センサの出力を差動 ADC に直接接続できます。ホール効果センサなどのシングルエンド信号の場合、シングルエンドアンプを使用できます。

その後、アンプの出力はオプションのフィルタに接続されます。フィルタは、測定する信号に応じて、多くの機能を持つことができます。いくつかの機能を示します。

- アンチエイリアシング用のローパスフィルタ
- 電源ノイズを減衰させるノッチフィルタ
- DC バイアスを除去し、信号の対象の帯域のみを抽出するバンドパスまたはハイパスフィルタ

次に、フィルタの出力が ADC に接続されます。MCU は、ADC からのデジタル化信号をさらに処理します。リソース制約の高い設計では、CM0+ などの MCU を使用してファームウェアによるフィルタを実装できます。

3.1 実装

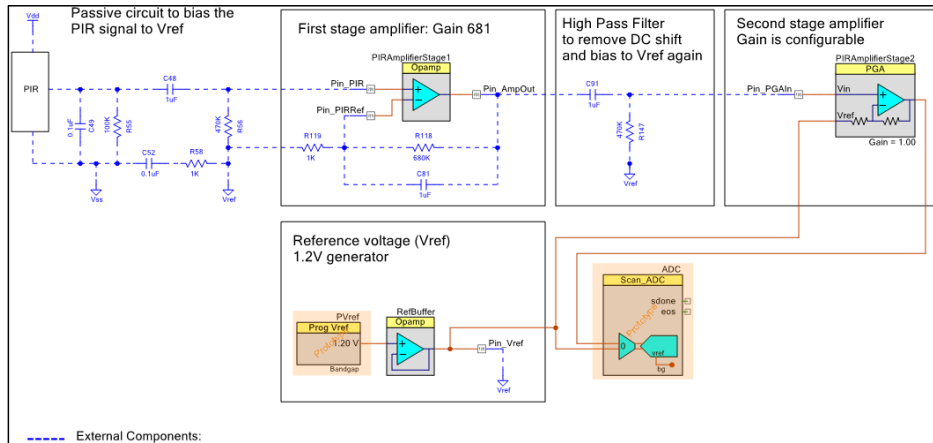
このセクションでは、電圧出力センサである PIR モーションセンサ用に AFE を実装する方法について説明します。ファームウェア実装の詳細については、サンプルコード [CE223618](#) を参照してください。

PIR モーションセンサは、センサ素子として赤外線に敏感な材料を使用します。それはソースフォロアモードの電界効果トランジスタ (FET) とともにパッケージ化されています。センサ素子の高インピーダンス出力をバッファするために FET が必要です。

センサ素子が赤外線にさらされると、素子の両端に電圧が発生します。これらの素子は、ひとつの素子によって生成された電圧が他の素子によって差し引かれるように配置されています。この配置は、共通の信号をキャンセルし、感知素子への入射赤外線レベルに差がある場合にのみ電圧を生成します。センサパッケージは、各素子に対して一意の視野を持つように設計されています。IR 放射源が視野を横切ると、センサは負荷抵抗の両端に差動信号を発生します。視野角を拡大するために、フレネルレンズが PIR モーションセンサに取り付けられています。感度が向上し、検出距離が向上します。

図 3 図 3 に、PIR モーションセンサの AFE として構成された PSoC 4100PS の PSoC Creator™ 回路図を示します。

図 3. PSoC Creator の回路図



センサ電圧パルスは、まずハイパスフィルタを通して、DC 電圧を次のステージに通さないようにします。ハイパスフィルタからのピークツーピーク電圧信号はミリボルトのオーダーであり、増幅する必要があります。3m の距離で人体の動きを検出するには、約 1000 のゲインを持つアンプが必要です。単段アンプでのこのような高ゲインは、アンプの出力を飽和させます。したがって、このような高いゲインでの増幅には 2 段アンプが最適です。総ゲインは 2 つのステージに分割されます。初段アンプは、内部オペアンプと外部ゲイン設定抵抗を使用して、非反転アンプ構成を使用します。2 段目のアンプは PGA コンポーネントを使用しています。PVref コンポーネントからの基準電圧は、Opamp コンポーネントを使用してバッファリング後、PIR センサおよびアンプ段にバイアスを与えるために使用されます。2 段目のアンプである PGA の出力は、Scan_ADC コンポーネントに接続されています。Scan_ADC の詳細については、[Scanning SAR ADC データシート](#)を参照してください。Scan_ADC の結果をしきい値と比較して、IR 放射オブジェクトの動きを検出します。

4 抵抗センサ用 AFE

抵抗センサでは、センサの抵抗は測定される物理量に応じて変化します。抵抗センサの例は次のとおりです。

- サーミスタおよび抵抗温度検出器 (RTD) センサでは、抵抗は温度によって変化します。
- ひずみゲージでは、負荷によって抵抗が変化します。
- 光依存抵抗器 (LDR) では、光の強さによって抵抗が変化します。

このセクションでは、抵抗測定の 2 つの方法について説明します。また、PSoC 4100PS での実装をサンプルコードとともに説明します。パフォーマンス分析セクションでは、抵抗測定方法の 1 つに関連する誤差と、そのような誤差を処理するためのヒントについて説明します。

4.1 電流源メソッド

この方法では、[図 4 図 4](#) に示すセンサ (R_{sensor}) は、定電流源を使用して励起されます。センサの電圧降下は、ADC を使用して測定されます。電圧が測定された後、[式 1 式 1](#) を使用してセンサの抵抗が計算されます。この方法では、抵抗測定の精度は、電流源の精度、ADC のオフセットおよびゲイン誤差に依存します。ADC ゲイン誤差は基準電圧の精度に依存することに注意してください。

Formatted: Superscript

Formatted: Font: (Default) MS Gothic, (Asian) MS Gothic, Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font: (Default) MS Gothic, (Asian) MS Gothic, Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

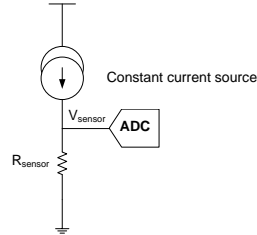
Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Superscript

$$R_{\text{sensor}} = \frac{V_{\text{sensor}}}{I} \quad \text{式 1}$$

図 4. 定電流源を使用した抵抗測定



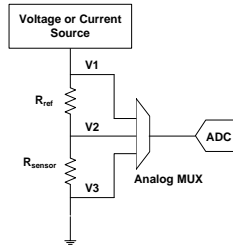
電流源メソッドの欠点は次のとおりです。

1. 現在のソースは非常に正確でなければなりません。電流誤差が発生すると、測定された抵抗に誤差が発生し、測定された物理特性に誤差が発生します。
2. 不正確な基準電圧によって引き起こされる ADC のオフセット、ゲイン、および積分非直線性 (INL) 誤差は、抵抗の測定値を不正確なものにする可能性があります。
3. R_{sensor} の非線形性は V_{sensor} の非線形性を引き起こし、物理特性の測定誤差を引き起こします。

4.2 レシオメトリック抵抗分割器法

図 5 図 6 に示す方法は、前述の方法に関連する誤差の一部を低減するために使用されます。基準抵抗器 (R_{ref}) とセンサ (R_{sensor}) は抵抗分割器を形成し、電圧または電流源を使用して励起されます。抵抗分割器は、必要に応じてデバイスの電源電圧から励起することもできます。 R_{sensor} は、基準抵抗とセンサの両端に発生する電圧の比率を使用して計算されます。

図 5. リファレンス抵抗を使用した抵抗測定



Note: R_{ref} と R_{sensor} の位置は、物理量の測定範囲に基づいて交換できます。たとえば、温度が測定対象の物理量であり、サーミスタのような負の温度係数素子を使用する場合は、サーミスタを上部に配置し、基準抵抗を下部に配置することをお勧めします。この構成により、高温でのサーミスタの抵抗が低くても、ADC 入力信号が測定範囲を下回ることはありません。

ADC を使用して V1、V2、および V3 を測定し、センサの抵抗を式 2 式 2 から計算します。

Formatted: Font: (Default) MS Gothic, (Asian) MS Gothic, Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

$$R_{sensor} = \frac{(V2 - V3)}{(V1 - V2)} * R_{ref} \quad \text{式 2}$$

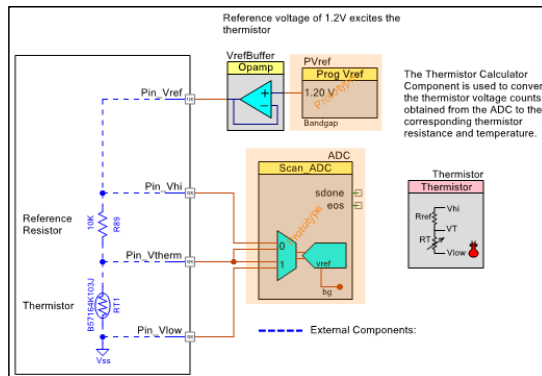
この方法では、センサ抵抗測定の精度は、リファレンス抵抗の精度と ADC の直線性にのみ依存します。この測定回路の精度を改善するには、正確な基準抵抗を使用するのが最も安価な方法です。この方法は、励起源の誤差、ADC オフセット、およびゲイン誤差とは無関係です。式 2 式 2 では、ADC のオフセット誤差は分子と分母の減算演算でキャンセルされ、ADC のゲイン誤差は分子と分母の比率でキャンセルされます。

4.2.1 抵抗分割法の実装

ここでは、サーミスタを使用して温度を測定するために、PSoC Creator で電位分割法を実装する方法について説明します。この例では、負の温度係数 (NTC) サーミスタ (B57164K103J) が使用されています。ここでは、AFE の設計と分析にのみ焦点を当てます。詳細については、サンプルコード [CE223621](#) を参照してください。

図 6 図 6 に、PSoC 4100PS で電位分割法がどのように実装されるかを示します。

図 6. PSoC Creator の温度測定用回路図



基準抵抗 R_1 はサーミスタと直列に接続されています。サーミスタは、プログラム可能なリファレンスコンポーネント PVref からのバッファ付き 1.2 V 電圧を使用してバイアスされます。リソース制約の高い設計では、デバイスの電源電圧を使用して、特定の温度範囲で測定の分解能を下げてサーミスタにバイアスをかけることができます。プログラマブルリファレンスコンポーネント PVref からのバイアス電圧を使用するもう 1 つの利点は、省電力設計で動的にオンまたはオフにできることです。

分圧器からの 3 つの電圧信号は、Scan_ADC コンポーネントの一部であるマルチプレクサに接続されます。3 つの電圧信号は 2 つの差動チャンネルに接続されます。Scan_ADC コンポーネントリファレンスは、ADC のフルスケール範囲を有効にするように設定されています。ADC のフルスケール範囲を使用すると、測定分解能が向上します。

ADC 変換出力は、Thermistor Calculator コンポーネントの Thermistor_GetResistance() API に渡され、サーミスタの抵抗値を計算します。次に、抵抗値が Thermistor_GetTemperature() API に渡され、ユーザーがサーミスタ計算コンポーネントに入力したサーミスタパラメータに基づいて温度が計算されます。温度は、Steinhart-Hart 法またはルックアップテーブル法のいずれかを使用して計算できます。サンプルコード [CE223621](#) は、ルックアップテーブル法を使用してコードを実装します。

同じ手法を使用して、RTD、ひずみゲージ、および LDR などの他のタイプのセンサの抵抗を測定できます。

Formatted: Superscript

Formatted: Font color: Custom Color (RGB (31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB (31, 82, 162)), (Asian) Japanese

Formatted: Font: (Default) MS Gothic, (Asian) MS Gothic, Font color: Custom Color (RGB (31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB (31, 82, 162)), (Asian) Japanese

4.3 精度分析

抵抗センサの測定精度は、抵抗測定の精度に依存します。抵抗を測定する場合、以下が誤差の原因です。

1. 基準抵抗器の許容差
2. ADC のオフセットとゲイン
3. ADC の非直線性

抵抗測定の誤差に基づいて物理量の誤差を計算するには、[温度や圧力などの物理量の誤差のセクション](#)を参照してください。

基準抵抗の許容差

以下に再現される [式 2 式 2](#) から始めましょう。

$$R_{sensor} = \frac{(V2 - V3)}{(V1 - V2)} * R_{ref}$$

ここで、誤差 ΔR_{ref} を R_{ref} に追加します。これにより、 R_{sensor} に誤差 ΔR_{sensor} が追加されます。

$$R_{sensor} + \Delta R_{sensor} = \frac{(V2 - V3)}{(V1 - V2)} * (R_{ref} + \Delta R_{ref}) \tag{式 3}$$

$(V2-V3)/(V1-V2)$ の変化は小さく、計算を簡単にするために無視されます。[式 3 式 3](#) を [式 2 式 2](#) で除算すると、[式 6 式 6](#) が得られます。

Formatted: Font color: Custom Color (RGB(31, 82, 162))

Formatted: Font color: Custom Color (RGB(31, 82, 162))

Formatted: Font color: Custom Color (RGB(31, 82, 162))

$$\frac{R_{sensor} + \Delta R_{sensor}}{R_{sensor}} \cong \frac{R_{ref} + \Delta R_{ref}}{R_{ref}} \tag{式 4}$$

$$1 + \frac{\Delta R_{sensor}}{R_{sensor}} \cong 1 + \frac{\Delta R_{ref}}{R_{ref}} \tag{式 5}$$

$$\frac{\Delta R_{sensor}}{R_{sensor}} \cong \frac{\Delta R_{ref}}{R_{ref}} \tag{式 6}$$

上記の式に見られるように、 R_{sensor} で導入された%誤差は、 R_{ref} での%誤差とほぼ同じです。 R_{ref} の%誤差は、抵抗の許容誤差です。たとえば、 R_{ref} によって導入される最大誤差は約 0.1% です。

[表 1 表 1](#) は、10 K のリファレンス抵抗の測定抵抗の実際の誤差を示しています。 R_{sensor} の測定値の%誤差は、 R_{sensor} のすべての値で同じです。 R_{ref} に 0.1% の誤差があると、 R_{sensor} に 0.1% の誤差が生じます。 R_{ref} で 5% の誤差が発生すると、 R_{sensor} で 4.76% の誤差が発生します。

Formatted: Font: (Default) MS Gothic, (Asian) MS Gothic, Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

表 1. 基準抵抗のさまざまな誤差値に対する抵抗測定の誤差

R _{sensor}	R _{ref} の誤差			
	0.10%	1.00%	2%	5%
10 KΩ	0.10%	0.99%	1.96%	4.76%

ADC オフセットおよびゲイン誤差

ここで、式 2 にオフセット誤差 V_{OS} とゲイン誤差 x を導入しましょう。

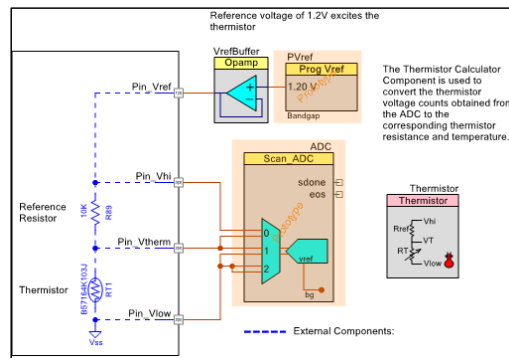
$$R_{sensor} = \frac{x(V2 + V_{OS}) - x(V3 + V_{OS})}{x(V1 + V_{OS}) - x(V2 + V_{OS})} * R_{ref} \quad \text{式 7}$$

$$R_{sensor} = \frac{x(V2 + V_{OS} - V3 - V_{OS})}{x(V1 + V_{OS} - V2 - V_{OS})} * R_{ref} \quad \text{式 8}$$

式 8 は、オフセットとゲインの両方の誤差が電位分割法でキャンセルされることを示します。したがって、ADC のオフセットとゲインの誤差は、測定の精度に影響を与えません。

図 6 に示すように差動測定を使用する場合、図 7 のチャンネル #2 に示すように、もう 1 つの差動チャンネルを追加し、両方の端子をグラウンドに短絡することにより、ADC のオフセット電圧を測定できます。次に、オフセットカウントを、channel #0 と channel #1 に対応するカウントから差し引いて、オフセット誤差を削除できます。

図 7. ADC の差動モードでのオフセット補償



ADC 非線形性誤差

PSoC 4100PS デバイスの SAR ADC の最大 INL は ± 2 LSB です。1.2 V の範囲と 12 ビットの分解能の場合、INL 誤差は約 1.2 mV です。INL 電圧が式 2 の分子と分母の両方に影響を与えると、最悪の場合の誤差が発生します。次の例について考えます。

$$R_{sensor} = 7 \text{ k}\Omega \quad R_{ref} = 10 \text{ k}\Omega \quad \text{励起電圧} = 1.2 \text{ V}$$

INL 誤差による測定された抵抗の誤差の最悪値は 10 で、0.14% の誤差です。R_{sensor} のさまざまな値の ADC INL 誤差による測定された抵抗の誤差の最悪値を図 8 に示します。

Formatted: Superscript

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font: (Default) MS Gothic, (Asian) MS Gothic, Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font: (Default) MS Gothic, (Asian) MS Gothic, Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

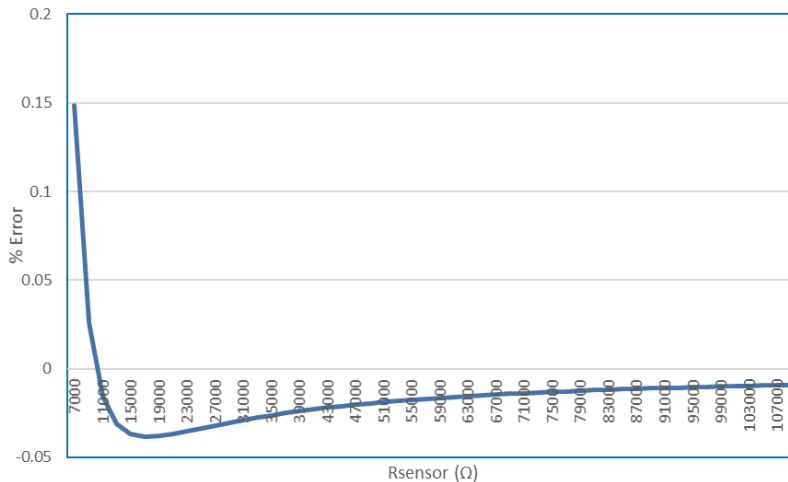
Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font: (Default) MS Gothic, (Asian) MS Gothic, Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

図 8. INL と R_{sensor} による抵抗の最悪の場合の誤差



温度や圧力などの物理量の誤差

ここでは、測定された抵抗値 (ΔR_{sensor}) の誤差による、測定された物理量の誤差 (温度の場合の ΔT など) を計算する正確な手順について説明します。

- 式 6 式 6 から、既知の R_{ref} 、 ΔR_{ref} 、および R_{sensor} のセットのセンサ抵抗の誤差 (ΔR_{sensor}) を計算します。
- 式 9 式 9 から、逸脱したセンサ抵抗 R'_{sensor} の絶対値を計算します。

$$R'_{\text{sensor}} = R_{\text{sensor}} \pm \Delta R_{\text{sensor}} \quad \text{式 9}$$

- センサの製造元のデータシートから、 R_{sensor} と R'_{sensor} に対応する物理量を見つけます。
- 手順 3 で計算された物理量 (温度や圧力など) の値の差を取って、 R_{ref} の変動による物理量の誤差を計算します。この値は、物理量の誤差に対応しています。

基準抵抗の選択

いずれかの極端な温度で最大の分解能を実現するには、基準抵抗をサーミスタの温度範囲の中央での抵抗に近づける必要があります。1 つの極端で温度を測定することにさらに関心がある場合、基準抵抗は、測定される極限の温度でのサーミスタの抵抗と一致する必要があります。

変換精度

抵抗を物理量に変換する精度は、センサの非線形特性に一致させるための曲線フィッティングの精度に依存します。たとえば、NTC (負の温度係数) サーミスタの場合、サーミスタの抵抗は温度が上昇するにつれて減少します。温度による抵抗の変化は非線形です。Thermistor Calculator コンポーネントでは、 $-40^{\circ}\text{C} \sim 125^{\circ}\text{C}$ の温度範囲にわたる変換誤差は 0.01°C 未満です。

Formatted: Font color: Custom Color (RGB (31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB (31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB (31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB (31, 82, 162)), (Asian) Japanese

5 静電容量センサ用 AFE

静電容量センサでは、センサ端子間の静電容量は測定される物理量に応じて変化します。静電容量センサの例には次のものがあります。

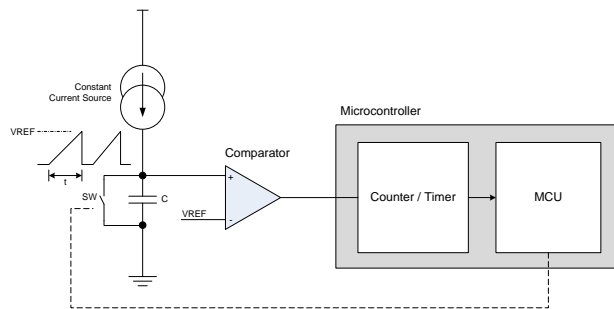
- 湿度センサ
- 圧力センサ
- 近接センサ

ここでは、静電容量センサを測定するための2つのAFE設計について説明します。また、PSoC 4100PSを使用した実装をサンプルコードとともに説明します。パフォーマンス分析セクションでは、静電容量測定方法の1つに関連する誤差と、そのような誤差を処理するためのヒントについて説明します。

5.1 単一勾配法

この方法では、定電流源を使用してコンデンサを充電します。コンデンサを既知の電圧に充電するのにかかる時間を使用して、静電容量を測定できます。

図 9. 静電容量を測定する単一勾配法



上記の回路では、マイクロコントローラは最初にスイッチ SW (GPIO または外部 FET の可能性があります) を閉じたままにして、コンデンサを放電状態に保持します。次に、マイクロコントローラがスイッチを解放します。電流源がコンデンサを充電し、コンデンサの両端の電圧が直線的に増加します。コンデンサ両端の電圧が V_{REF} より大きくなると、コンパレータ出力がハイになります。マイクロコントローラはタイマー/カウンタを使用してこの時間を測定し、[式 10 式 10](#) を使用して静電容量 C を計算します。

$$C = \frac{I t}{V} \quad \text{式 10}$$

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

ここでは、以下のとおりです。

I = 定電流源からの電流

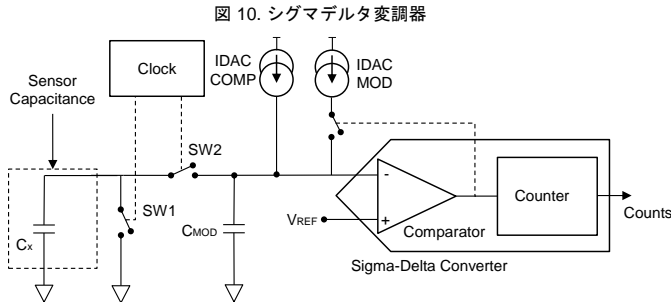
V = コンパレータの基準電圧

t = コンデンサを V_{REF} まで充電するのにかかる時間

この方法の精度は、定電流源、コンパレータの基準電圧、およびタイマーの精度に依存します。

5.2 シグマデルタ変調方式

この方法では、シグマデルタ変調器が静電容量をデジタルビットストリームに変換します。ビットストリームの密度は静電容量値に正比例します。図 10 図 10 に、シグマデルタ変調器を示します。

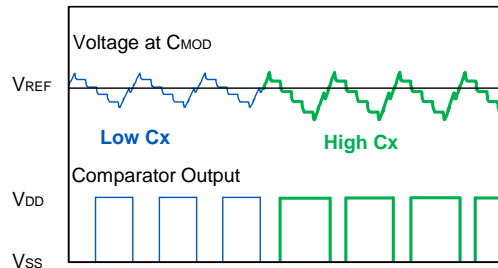


IDAC_{MOD} は、変調コンデンサ C_{MOD} を充電する電流源です。C_{MOD} の両端の電圧が V_{REF} を超えると、コンパレータの出力が Low になり、IDAC_{MOD} をオフにします。

C_x は測定するセンサの静電容量です。SW1 と SW2 は C_x とともにスイッチコンデンサセルを形成します。SW1 と SW2 は、クロックソースによって位相がずれて駆動されます。SW2 がオンのとき、電荷は C_{MOD} から C_x に転送されます。SW1 がオンのとき、C_x は放電されます。したがって、C_x、SW1、および SW2 は、C_{MOD} を放電する抵抗として機能します。C_{MOD} が放電すると、電圧が低下します。電圧が V_{REF} を下回ると、コンパレータの出力がハイになり、IDAC_{MOD} がオンになり、充電サイクルが再開されます。IDAC_{COMP} は、IDAC_{MOD} と併用して測定のダイナミックレンジを拡大できる固定電流源です。

図 11 図 11 に、コンパレータの出力と C_{MOD} の両端の電圧を示します。

図 11. 異なる C_x 値に対するコンパレータの出力



C_x が小さい (信号が青色で示されている) と、C_{MOD} から取り出される電荷が少なくなるため、C_{MOD} は V_{REF} までより速く充電されます。これにより、コンパレータの出力が HIGH に留まる時間が短くなります。C_x が大きいほど (信号は緑色で表示)、C_{MOD} から取り出される電荷が増えるため、C_{MOD} の充電時間が長くなります。これにより、コンパレータの出力が HIGH に留まる時間が長くなります。カウンターは、一定期間のコンパレータの出力を積分し、カウンターの出力は静電容量に直接比例します。

Formatted: Superscript

Formatted: Font: (Default) MS Gothic, (Asian) MS Gothic, Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font: (Default) MS Gothic, (Asian) MS Gothic, Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

次の式は、「raw count」とも呼ばれるカウンターの出力と静電容量の関係を示します。

$$\text{raw count} = (2^N - 1) \frac{(V_{DD} - V_{REF}) F_{SW}}{I_{MOD}} C_S - (2^N - 1) \frac{I_{COMP}}{I_{MOD}} \quad \text{式 11}$$

ここでは、以下のとおりです。

C_S - 測定中の静電容量

N - ビット単位のカウンター分解能

F_{SW} - SW1 と SW2 が動作する周波数

I_{MOD} - IDAC_{MOD} からの電流

I_{COMP} - IDAC_{COMP} からの電流

V_{REF} - コンパレータの基準電圧

式 11 式 11 からわかるように、測定の精度は IDAC_{MOD}、IDAC_{COMP}、 V_{DD} 、および V_{REF} に依存します。これらのパラメータのいずれかに誤差があると、測定された静電容量に誤差が生じます。

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

これらのパラメータの誤差を補償するために、良好な許容範囲と温度安定性を備えた単一のリファレンス コンデンサが使用されます。センサ コンデンサとリファレンス コンデンサの両方がシグマデルタ変調器で測定され、センサ コンデンサの値は raw count の比率から見つけることができます。

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

$$C_{\text{sensor}} = \frac{C_{\text{ref}}}{\text{RawCounts}_{C_{\text{ref}}}} * \text{RawCounts}_{C_{\text{sensor}}} \quad \text{式 12}$$

この方法は単一勾配法よりもはるかに複雑に見えますが、この方法を実装するために必要なすべてのハードウェアは、CapSense® シグマデルタ (CSD) ブロックの PSoC 4100PS 内にあります。これにより、この手法を簡単に使用できます。

5.2.1 寄生容量

センサの静電容量を検出ハードウェアに接続する PCB トレースの寄生静電容量とピンの静電容量が結果に追加され、オフセット誤差が発生します。

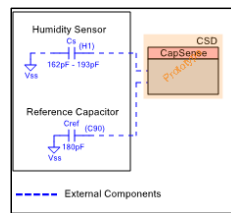
PCB トレースによって引き起こされるこのオフセットに対処するには 2 つの方法があります。1 つ目は、PCB トレースをできるだけ短くして寄生容量を減らすことです。この方法は、基準コンデンサ (C_{ref}) に使用されます。これは通常、センサ コンデンサには実用的ではありません。このような場合、システムの寄生容量は、センサが PSoC から切断された状態で初期測定を行うことによって測定されます。その後、センサ コンデンサの測定値から結果を差し引くことができます。

5.2.2 シグマデルタ法の実装

ここでは、PSoC 4100PS を使用して湿度センサの静電容量を測定するためにシグマデルタ変調方式を使用する方法について説明します。ファームウェア設計の詳細については、サンプルコード [CE223620](#) を参照してください。

[図 12 図 12](#)に、湿度センサプロジェクトの PSoC Creator の回路図を示します。CapSense コンポーネントは、シグマデルタ変調器ハードウェアを実装します。コンポーネントには、静電容量に対応する raw カウントを報告する API 関数もあります。CapSense コンポーネントは通常、静電容量ボタン、スライダ、およびトラックパッドなどのタッチセンシングアプリケーションに使用されます。この例では、CapSense コンポーネントは湿度センサの静電容量を測定するために使用されます。

図 12. PSoC 4 の CSD 回路図



CapSense コンポーネントを使用して、PSoC 4100PS は基準コンデンサ値 (C_{ref}) に対応する raw カウント ($RawCount_{CS}$) を測定します。次に、センサが接続されていない raw counts が測定されます。これらの raw counts ($RawCount_{Cos}$) は、寄生容量 (C_{os}) に相当しています。基準コンデンサ (C_{ref}) は既知であるため、寄生容量の値は [式 12 式 12](#) を使用して求めることができます。センサが接続された raw counts ($RawCount_{CS}$) が測定されます。センサコンデンサの値は、[式 13 式 13](#) を使用して計算されます。

$$C_s = \frac{C_{ref} * (RawCount_{CS} - RawCount_{Cos})}{RawCount_{Cref} - RawCount_{Cos}} \quad \text{式 13}$$

ここでは、以下のとおりです。

C_{ref} – 基準容量値

C_s – センサ容量値

$RawCount_{CS}$ – 湿度センサの静電容量の測定からの raw counts

$RawCount_{Cos}$ – Cos に対応する raw counts 値 (PCB トレースと GPIO キャパシタンスが原因)

$RawCount_{Cref}$ – 基準静電容量の測定からの raw counts

センサの静電容量が計算されると、[式 14 式 14](#) を使用して湿度が計算されます。

$$\%RH = \frac{C_s - C_{nom}}{Sensitivity} + \%RH_{nom} \quad \text{式 14}$$

ここでは、以下のとおりです。

$\%RH$ – 相対湿度

C_s – 湿度センサの測定静電容量

C_{nom} – センサの公称静電容量 (HS1100 の場合、これは 180pF)

$\%RH_{nom}$ – センサの公称湿度 (HS1100 の場合、これは 55%RH)

Formatted: Superscript

Formatted: Font: (Default) MS Gothic, (Asian) MS Gothic, Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB(31, 82, 162)), (Asian) Japanese

感度 – センサの pF/%RH 値 (HS1100 の場合、これは 0.34pF/%RH)

5.3 精度分析

湿度測定 システム精度は、CapSense コンポーネントによる静電容量測定精度とセンサの直線性に依存します。

前のセクションで見たように、静電容量は式 15 式 15 を使用して計算されます。

$$C_s = \frac{C_{ref} * (RawCount_{CS} - RawCount_{cos})}{RawCount_{Cref} - RawCount_{cos}} \quad \text{式 15}$$

式 15 式 15 から、静電容量測定精度は、基準コンデンサ C_{ref} の精度に依存します。C_{ref} の変化による測定された静電容量 C の誤差は、式 16 式 16 で与えられます。

$$Error = \Delta C = \frac{\pm C * \Delta C_{ref}}{C_{ref} \pm \Delta C_{ref}} \quad \text{式 16}$$

6 まとめ

このアプリケーションノートでは、電圧、抵抗、および容量センサの AFE について説明します。PSoC Creator コンポーネントを使用して、AFE の各要素の実装の詳細を提供します。

7 関連資料

アプリケーションノート	
AN79953 - PSoC [®] 4 入門	PSoC 4 を紹介し、PSoC 4 アーキテクチャと開発ツールの探索を支援し、PSoC Creator を使用して最初のプロジェクトを作成する方法を示します。
AN86233 - PSoC [®] 4 低消費電力モードおよび消費電力削減技術	PSoC 4 低電力モードと機能を使用して、非常に低い電力レベルで動作する方法について説明します。
AN88619 - PSoC 4 Hardware Design Consideration	PSoC 4 デバイスを中心にハードウェアシステムを設計する方法を示します。
AN73854 - PSoC [®] 3, PSoC 4, and PSoC 5LP Introduction to Bootloaders	ブートローダの理論とテクノロジーを簡単に紹介します。
AN89056 - PSoC [®] 4 – IEC 60730 Class B and IEC 61508 SIL Safety Software Library	PSoC 4 IEC 60730 Class B および IEC 61508 Safety Integrity Level (SIL) Safety Software Library とサンプルプロジェクトについて説明します。
AN60590 - PSoC [®] 3, PSoC 4, and PSoC 5LP – Temperature Measurement with a Diode	PSoC 3, PSoC 4、および PSoC 5LP を使用したダイオードベースの温度測定について説明します。
AN70698 - PSoC [®] 3, PSoC 4, and PSoC 5LP – Temperature Measurement with RTD	RTD を使用した温度測定の理論を説明し、RTD を PSoC とインターフェースするプロジェクトを含みます。
AN66477 - PSoC [®] 3, PSoC 4, and PSoC 5LP – Temperature Measurement with a Thermistor	PSoC 3, PSoC 4、または PSoC 5LP を使用してサーミスタで温度を測定する方法を説明します。
AN89610 - PSoC [®] 4 and PSoC 5LP Arm [®] Cortex [®] Code Optimization	PSoC 4 および PSoC 5LP の Arm Cortex CPU 用に C およびアセンブラコードを最適化する方法を示します。
AN90799 - PSoC [®] 4 Interrupts	PSoC 4 の割り込みアーキテクチャと PSoC Creator の構成について説明します。
AN86439 - PSoC [®] 4 - GPIO ピンの使用	PSoC 4 GPIO ピンを効果的に使用してさまざまなユースケース例を示し、その機能を説明する方法を説明します。

Formatted: Superscript

Formatted: Font color: Custom Color (RGB (31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB (31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB (31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB (31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB (31, 82, 162)), (Asian) Japanese

Formatted: Font color: Custom Color (RGB (31, 82, 162)), (Asian) Japanese

AN85951 - PSoC 4 CapSense® デザインガイド	PSoC 4 で容量性タッチセンシングアプリケーションを設計する方法を示します。
AN92239 - Proximity Sensing with CapSense®	CapSense を使用した PSoC 4 近接センシングのデモを示します。
AN86526 - PSoC 4 I ² C ブートローダ	PSoC 4 の I ² C ベースのブートローダについて説明します。
AN68272 - PSoC [®] 3、PSoC 4、PSoC 5LP、および PSoC アナログコプロセッサ UART ブートローダ	PSoC 3、PSoC 4、PSoC 5LP、および PSoC アナログコプロセッサ用の UART ベースのブートローダについて説明します。
AN87391 - PSoC [®] 4 セグメント LCD の直接駆動	PSoC 4 に統合された LCD ドライバーを使用してセグメント LCD ガラスを駆動することがいかに簡単かを示します。
AN84858 - 外部マイクロコントローラを使用した PSoC [®] 4 プログラミング (HSSP)	モジュラーCコードを使用して、外部マイクロコントローラで PSoC 4 デバイスプログラミングを実装する方法を示します。
サンプルコード	
CE223618 - Interfacing PSoC 4 with PIR Motion Sensor	PSoC 4 を PIR モーションセンサとインターフェースする方法を示します。
CE223621 - Interfacing PSoC 4 with Temperature Sensor	PSoC 4 をサーミスタとインターフェースする方法を示します。
CE223620 - Interfacing PSoC 4 with Humidity Sensor	PSoC 4 を湿度センサとインターフェースする方法を示します。

著者について

名前: Dineshababu Mani
 役職: スタッフ アプリケーション エンジニア
 経歴: Dineshababu は、インドのピラニにある BITS のマイクロエレクトロニクスの修士号を持っています。彼の関心領域には、混合信号システム設計、制御システム、信号処理が含まれます。

改訂履歴

文書名: AN223616 – PSoC 4 を使用した AFE 実装

文書番号: 002-xxxxx

版数	変更内容
**	本版は英語版 002-23616 Rev. **について、CYPRESS DEVELOPER COMMUNITYの参画者によって日本語に翻訳されたドキュメントです。

セールス, ソリューションおよび法律情報

ワールドワイドな販売と設計サポート

サイプレスは、事業所、ソリューションセンター、メーカー代理店、および販売代理店の世界的なネットワークを保持しています。お客様の最寄りのオフィスについては、[サイプレスのロケーションページ](#)をご覧ください。

製品

Arm® Cortex® Microcontrollers	cypress.com/arm
車載用	cypress.com/automotive
クロック&バッファ	cypress.com/clocks
インターフェース	cypress.com/interface
IoT (モノのインターネット)	cypress.com/iot
メモリ	cypress.com/memory
マイクロコントローラ	cypress.com/mcu
PSoC	cypress.com/psoc
電源用 IC	cypress.com/pmic
タッチセンシング	cypress.com/touch
USB コントローラー	cypress.com/usb
ワイヤレス	cypress.com/wireless

PSoC®ソリューション

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6 MCU](#)

サイプレス開発者コミュニティ

[コミュニティ](#) | [サンプルコード](#) | [Projects](#) | [ビデオ](#) | [ブログ](#) | [トレーニング](#) | [Components](#)

テクニカルサポート

cypress.com/support

本書で言及するその他のすべての商標または登録商標は、それぞれの所有者に帰属します。



© Cypress Semiconductor Corporation, 2018-2020. 本書面は、Cypress Semiconductor Corporation 及び Spansion LLC を含むその子会社 (以下「Cypress」という。) に帰属する財産である。本書面 (本書面に含まれ又は言及されているあらゆるソフトウェア若しくはファームウェア (以下「本ソフトウェア」という。)) を含むは、アメリカ合衆国及び世界のその他の国における知的財産法令及び条約に基づき Cypress が所有する。Cypress はこれらの法令及び条約に基づく全ての権利を留保し、本段落で特に記載されているものを除き、その特許権、著作権、商標権又はその他の知的財産権のライセンスを一切許諾しない。本ソフトウェアにライセンス契約書が伴っており、かつ Cypress との間で別途本ソフトウェアの使用方法を定める書面による合意がない場合、Cypress は、(1) 本ソフトウェアの著作権に基づき、(a) ソースコード形式で提供されている本ソフトウェアについて、Cypress ハードウェア製品と共に用いるためにのみ、かつ組織内部でのみ、本ソフトウェアの修正及び複製を行うこと、並びに (b) Cypress のハードウェア製品ユニットに用いるためにのみ、(直接又は再販売者及び販売代理店を介して間接のいずれかで) 本ソフトウェアをバイナリーコード形式で外部エンドユーザーに配布すること、並びに (2) 本ソフトウェア (Cypress により提供され、修正がなされていないもの) が抵触する Cypress の特許権のクレームに基づき、Cypress ハードウェア製品と共に用いるためにのみ、本ソフトウェアの作成、利用、配布及び輸入を行うことについての非独占的譲渡不能一身専属的ライセンス (サブライセンスの権利を除く) を付与する。本ソフトウェアのその他の使用、複製、修正、変換又はコンパイルを禁止する。

適用される法律により許される範囲内で、Cypress は、本書面又はいかなる本ソフトウェア若しくはこれに伴うハードウェアに関しても、明示又は黙示を問わず、いかなる保証 (商品性及び特定の目的への適合性の黙示の保証を含むがこれらに限られない) も行わない。いかなるコンピューティングデバイスも絶対に安全ということはない。従って、Cypress のハードウェアまたはソフトウェア製品に講じられたセキュリティ対策にもかかわらず、Cypress は、Cypress 製品への権限のないアクセスまたは使用といったセキュリティ違反から生じる一切の責任を負わない。加えて、本書面に記載された製品には、エラーと呼ばれる設計上の欠陥またはエラーが含まれている可能性があり、公表された仕様とは異なる動作をする場合がある。適用される法律により許される範囲内で、Cypress は、別途通知することなく、本書面を変更する権利を留保する。Cypress は、本書面に記載のある、いかなる製品若しくは回路の適用又は使用から生じる一切の責任を負わない。本書面で提供されたあらゆる情報 (あらゆるサンプルデザイン情報又はプログラムコードを含む) は、参照目的のためのみに提供されたものである。この情報で構成するあらゆるアプリケーション及びその結果としてのあらゆる製品の機能性及び安全性を適切に設計、プログラム、かつテストすることは、本書面のユーザーの責任において行われるものとする。Cypress 製品は、兵器、兵器システム、原子力施設、生命維持装置若しくは生命維持システム、蘇生用の設備及び外科的移植を含むその他の医療機器若しくは医療システム、汚染管理若しくは有害物質管理の運用のために設計され若しくは意図されたシステムの重要な構成部分としての使用、又は装置若しくはシステムの不具合が人身傷害、死亡若しくは物的損害を生じさせるようなその他の使用 (以下「本目的外使用」という。) のためには設計、意図又は承認されていない。重要な構成部分とは、その不具合が装置若しくはシステムの不具合を生じさせるか又はその安全性若しくは実効性に影響すると合理的に予想できるような装置若しくはシステムのあらゆる構成部分をいう。Cypress 製品のあらゆる本目的外使用から生じ、若しくは本目的外使用に関連するいかなる請求、損害又はその他の責任についても、Cypress はその全部又は一部を問わず一切の責任を負わず、かつ Cypress はそれら一切から本書により免除される。Cypress は Cypress 製品の目的外使用から生じ又は本目的外使用に関連するあらゆる請求、費用、損害及びその他の責任 (人身傷害又は死亡に基づく請求を含む) から免責補償される。

Cypress、Cypress のロゴ、Spansion、Spansion のロゴ及びこれらの組み合わせ、WICED、PSoC、CapSense、EZ-USB、F-RAM、及び Traveo は、米国及びその他の国における Cypress の商標又は登録商標である。Cypress のより完全な商標のリストは、cypress.com を参照すること。その他の名称及びブランドは、それぞれの権利者の財産として権利主張がなされている可能性がある。