AN210998

PSoC 4 低消費電力 CapSense 設計

著者: Vijay K M

関連プロジェクト: あり

関連製品ファミリ: PSoC® 4 シリーズ

ソフトウェア バージョン: PSoC Creator™ 4.2

関連アプリケーション ノート: 完全な一覧については、こちらをクリックしてください。

その他のサンプルコードが必要な場合は、以下のとおり対応します。

ModusToolbox™ IDE または PSoC® Creator™を使用して、増え続ける数百のサンプルコードのリストにアクセスするには、GitHub リポジトリにアクセスしてください。ここでサイプレスビデオトレーニングライブラリを探索することもできます。

AN210998 は PSoC 4 デバイスを使用して低消費電力 CapSense® アプリケーションを設計する方法について説明します。また、CapSense アプリケーションの電力消費量計算の方法を説明し、電力消費量を最小化するためにハードウェア、ファームウェア、およびシステム レベル ガイドラインを提供します。

目次

1 はじめに

静電容量センスに基づく設計は幅広い電池駆動および携帯電子機器でますます多く採用されています。電池寿命が最も重要であるこれらの携帯機器の設計で、電力消費量はキーとなるパラメーターです。本アプリケーション ノートは PSoC 4 デバイスを使用した低消費電力 CapSense アプリケーションの設計方法について説明します。そのために、CapSense システムの電力消費量に影響する要素および低消費電力設計のためのハードウェア、ファームウェア、およびシステム レベル注意事項について説明します。

本書は読者が PSoC 4 アーキテクチャ、PSoC 4 CapSense、および PSoC Creator™ 統合開発環境 (IDE) を使用した PSoC 4 のアプリケーション開発に精通していることを前提としています。PSoC 4 の導入は、AN79953 – Getting Started with PSoC 4 を参照してください。PSoC 4 CapSense の初心者である場合、AN64846 – Getting Started with CapSense および AN85951 – PSoC 4 CapSense Design Guide を参照してください。PSoC Creator が初めての方は、PSoC Creator ホームページを参照してください。

本アプリケーション ノートの初めの節は電力消費量に影響する要素について説明します。後の節では低消費電力設計のためのハードウェア、ファームウェアやシステム レベルの注意事項について説明します。最後に、低消費電力の CapSense 設計を達成する方法について説明するサンプル プロジェクトを示します。

2 低消費電力設計への導入

ここではシステムの定義における許容電力の要件の重要性について説明します。次に CapSense コントローラーの電力消費量に影響する要素 (アクティブ時間、スリープ時間、動作電圧、動作周波数、ペリフェラルおよび GPIO 活性など) の概要を説明します。この説明はアクティブ時間に影響する要素および瞬間的な電流に影響する要素に分けられます。

2.1 許容電力の要件

許容電力の要件は、低消費電力の静電容量センス システムのシステム定義において主な要素の一つです。最初の段階で目標電力消費量を定義することは、システム設計者にハードウェアやファームウェアの開発のためのスタート ポイントを提供します。

許容電力は電池容量および電池寿命から導かれます。電池容量は物理的なフォーム ファクタ、コストおよびサイズに応じて決められます。許容電力を計算するために、設計者は電池寿命 (すなわち、システムが電池の交換・再充電無しに動作できる時間) を知る必要があります。電池寿命の要件はアプリケーションによって異なります。モバイル／ポータブル ガジェットなどの一部のアプリケーションに対し、電池寿命は数日間維持する必要があります。また、密閉される、または遠隔地に設置されるシステムは、電池寿命が数ヶ月または数年維持することを必要とします。このようなシステムでは電池交換が難しく、許容電力の要件は厳しくなります。所要電池寿命および最大電池容量 (システムで使用できる) が決められると、平均電流を計算できます。電池容量の詳細については選択された電池のデータシートを参照してください。

電池駆動の組込みシステムの平均電流 (許容電力) は以下の式を使って計算できます。

$$\text{最大平均電流} = \frac{Battery\ Capacity[mA{-}H]}{Required\ Battery\ Life\ [H]}$$

システム タスクはシステムの合計平均電流が許容電力により指定された最大平均電流より低くなるように設計する必要があります。電池駆動アプリケーションにおいて電池寿命を延ばすためには、平均電力消費量は重要です。

2.2 平均電流に影響する要素

CapSense システムの電力消費量を決定付ける主要素は、デバイスのアクティブ モードとインアクティブ モードでの時間の割合、およびアクティブ モードとインアクティブ モードでの電流です。アクティブ モードでは CapSense ハードウェアはセンサーをスキャンして、人間との相互作用を検出します。インアクティブ モードでは、デバイスはスリープ モードに移行します。通常のアプリケーションにおいて CapSense コントローラーは常にアクティブ状態である必要はありません。CapSense ハードウェアはセンサーを周期的にスキャンします。センサーがスキャンされている時、デバイスはアクティブ モードのままですが、スキャンされていない時、デバイスはインアクティブ モードになります。

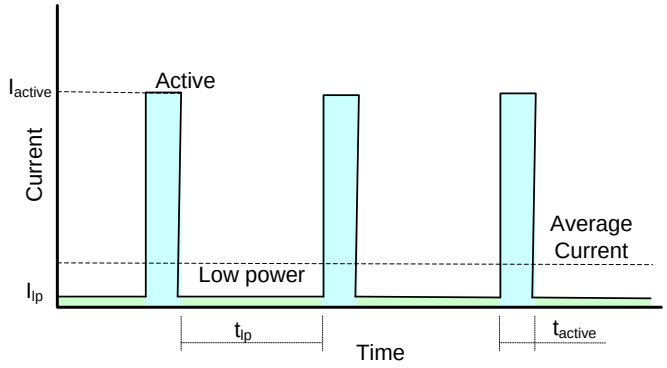
PSoC 4 は低消費電力モードをサポートし、デバイスの平均電力消費量を低減します。平均電力消費を低くするために、デバイスはインアクティブ モードの間最低消費電力モード (ディープスリープ) に移行させます。いくつかの PSoC 4 デバイスはディープスリープ モードより少ない消費電力のハイバネート モードとストップ モードをサポートしています。しかし、デバイスが割込みによってウェイクアップされる場合、これらのモードはソフトウェア リセットを引き起こし、そのリセットは CapSense を使用する場合に特別な処理を必要とします。したがって、ほとんどのシステムではディープスリープ モードの使用を推奨します。

CapSense ハードウェアがディープスリープ モードで無効にされるため、デバイスは周期的にウェイクアップして人間との相互作用をスキャンする必要があります。PSoC 4 内でのウォッチドッグ タイマー (WDT) を使用して、デバイスを周期的にディープスリープ モードからウェイクアップできます。

CapSense スキャンの間 (アクティブ モード) にも、スキャンの開始と終了の間で CPU の介入が必要とされないため、デバイスをスリープ モードに移行させられます。ファームウェアにスキャンの完了を待つタスク以外に追加のタスクがない場合、節電するために、スキャンを開始した後にデバイスをスリープ モードに移行させられます。CapSense ハードウェアはスキャンを完了した後、割込みを生成して、CPU をスリープ モードからウェイクアップし、アクティブ モードに復帰させます。

低い平均消費電力のために最適化されたシステムは、信頼性の高い動作を確保しつつ、ほとんどの時間で低消費電力モードになります。図 1 および式 1 は平均電流とアクティブ時間との関係を示します。

図 1. 一般的な低消費電力システムのタイミング図



デバイスが低消費電力モードから周期的に復帰してから、一定の時間でアクティブ モードにある場合、デバイスの平均消費電流を以下のとおりに計算できます。

$$\text{平均電流}(I_{AVE}) = \frac{((t_{active} * I_{active}) + (t_{lp} * I_{lp}))}{t_{active} + t_{lp}} \qquad \text{式 1}$$

t_{active} = アクティブ モードにある時間

I_{active} = アクティブ モード電流

t_{lp} = 低消費電力モードにある時間

I_{lp} = 低消費電力モードでの電流

デバイスの平均消費電力は、次のように計算します。

$$P_{AVE} = V_{DD} \times I_{AVE} \qquad \text{式 2}$$

式 2 に示すように、 電力消費量は動作電圧および平均電流に依存します。動作電圧は電力消費量に影響する主な要素です。電力消費量は動作電圧に正比例します。したがって、できるだけデバイスの動作電圧を低くすることを考慮してください。

式 1 に示すように、平均電流消費量はアクティブ時間と (アクティブ モードおよび低消費電力モードでの) 電流の低減により削減できます。

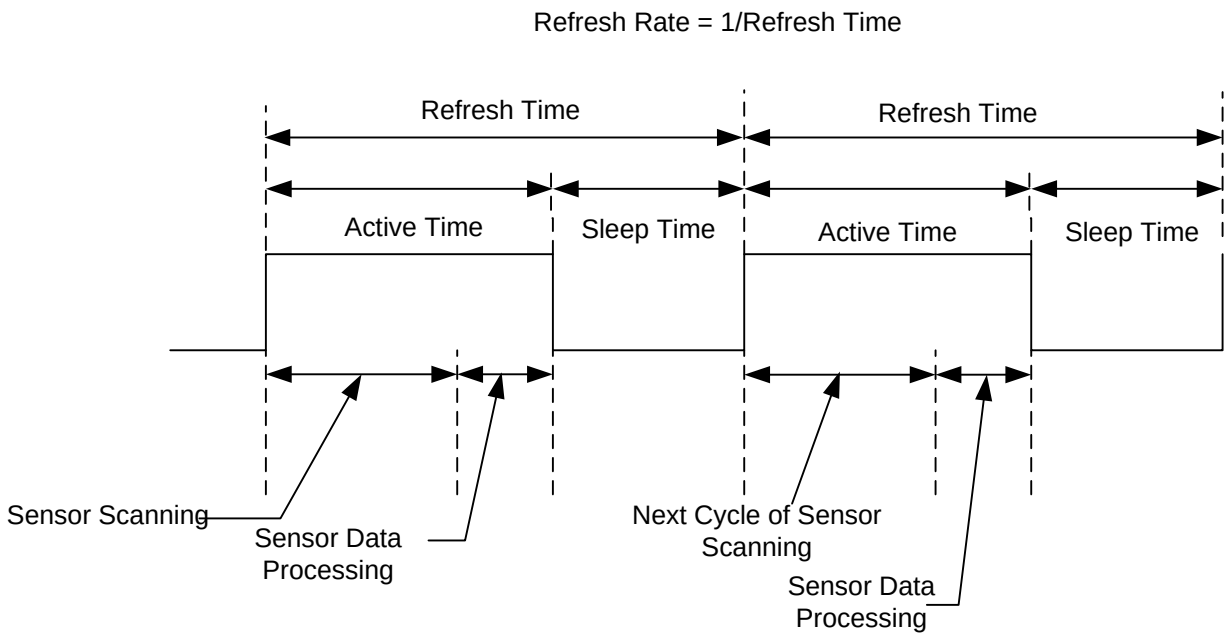
2.2.1 アクティブ時間に影響する要素

以下の要素はアクティブ時間に影響を与え、したがって全体的な電力消費量に影響を与えます。

- **センサー数:** センサーの数が多いほどセンサーの合計スキャン時間は大きくなります。固定した リフレッシュ レートに対して、センサーのスキャン時間が増えると、デバイスがスリープ モードにある期間は少なくなります。これにより、デバイスの消費電力が増加します。

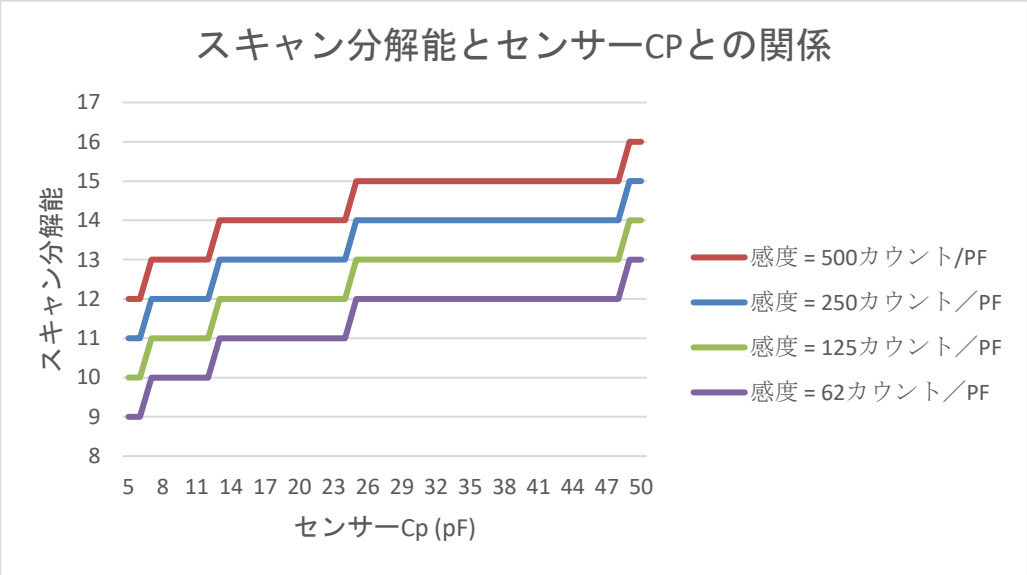
- **リフレッシュ レート:** リフレッシュ レートは 1 秒あたりのアクティブ-スリープ サイクルの数です。図 2 に示すように、このパラメーターは一定のアクティブ時間に対してデバイスがスリープ モードにある期間に影響をおよぼします。式 1 に示すように、平均の電力消費量はI_{AVE}の低減により削減できます。I_{AVE}はスリープ時間を増加することにより低減できます。リフレッシュ レートが低いほど、消費電流が少なくなります。しかし、低いリフレッシュ レートは応答時間を遅くします。リフレッシュ レートが長すぎると、エンドユーザーにとっての性能は低下します。そのため、応答時間と消費電力がトレードオフの関係になります。一般的な経験則として、リフレッシュ レートを最小 50Hz にして速い指タッチを検出した場合に、満足できるエンドユーザーにとっての性能が実現されます。リフレッシュ レートは、ユーザーがセンサーに触れているか、または長い時間触れていないかに基づきリフレッシュ レートを動的に変更することにより、さらに減らすことが可能です。この場合、ユーザーがセンサーに触れていなければ、センサーがより低いリフレッシュ レートでスキャンされます。ユーザーがセンサーを触れていると、速い指のタップを検出するためにリフレッシュ レートを増加させます。サンプル プロジェクトでは、応答時間の最適化および電力消費量の削減の方法を示します。

図 2. リフレッシュ レートとスリープ時間との関係



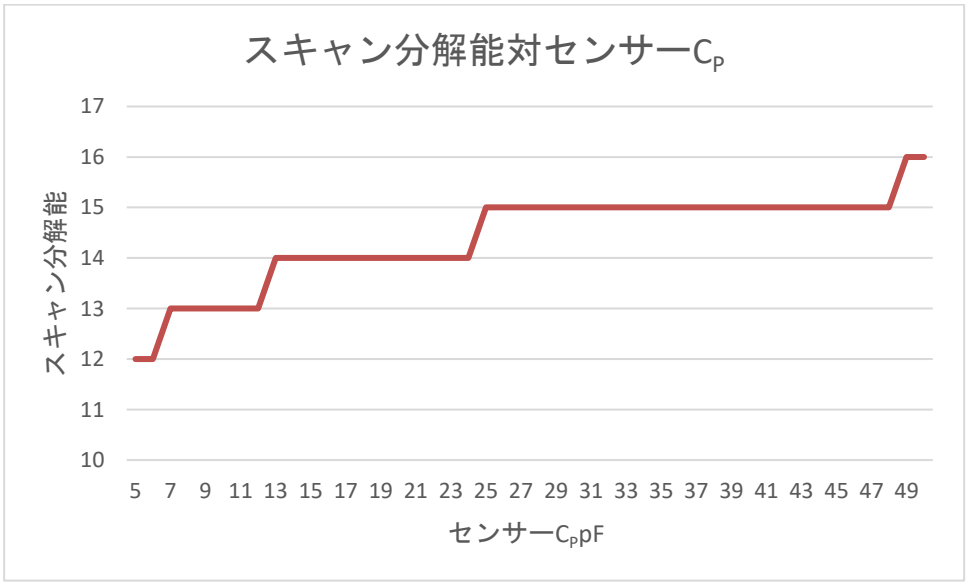
- **スキャン分解能／感度:** スキャン分解能は raw カウント カウンターの分解能です。感度パラメーターはスキャン分解能パラメーターと同様であり、指の静電容量 (C_F) の 1pF あたり達成される信号カウントとして定義されます。これらの要素の詳細については、ファームウェア設計上の注意事項節を参照してください。

図 3. 異なる感度値によるセンサー スキャン分解能とセンサーCP 値との関係



- **センサーの寄生容量 (CP):** センサーの CP はセンサーのスキャン時間に直接影響をおよぼします。図 4 はセンススキャン分解能パラメーターへのセンサーCP の影響を示します。大きい CP 値の場合、小さな CP 値のあるセンサーに比べ、センサーは所要信号を達成するために、より高いスキャン分解能でスキャンされる必要があります。スキャン分解能が高くなると、センススキャン時間が (固定の変調器クロックに対して) 長くなり、したがって、アクティブ時間が増加し、平均電力消費量が増加します。そのため、センサーの CP を以下のレイアウトのベスト プラクティスにより最小化します。本アプリケーション ノートの低消費電力向けのハードウェア設計上の注意事項の節はセンサーCP を最小化するガイドラインを提供します。

図 4. 500 カウント/pF の感度のための、センサー スキャン分解能とセンサーCP との関係



- **近接距離:** 大きい接近センシング距離を達成するために、近接センサーは高いスキャン分解能 (15 ビットまたは 16 ビット) でセンサーをスキャンする必要があります。分解能が高ければ高いほど、スキャン時間が長くなり、デバイスのアクティブ時間も長くなって、消費電力が増えます。したがって、近接センシング距離が大きければ大きいほど、必要とする消費電力が高くなります。

2.2.2 アクティブ モードおよび低消費電力モードでの電流に影響する要素

平均電流はアクティブ モードおよび低消費電力モードの瞬間的な電流消費量に依存します。以下の要素はこれらのモードでの瞬間的な電流に影響をおよぼします。

- **動作周波数:** アクティブ モードでのデバイスの電力消費量は動作周波数に正比例します。スキャンが完了した後、CPU は CapSense データを処理するため、動作周波数が高くなると処理完了時間が短くなります。しかし、高速のデータ処理を必要としないシステムに対しては、CPU 周波数をより低い値に減らし、CPU のアクティブ時間での電力消費量を削減させます。したがって、異なる周波数にまたがってアクティブ時間とアクティブ電流のトレードオフがあります。
- **未使用のペリフェラル:** ペリフェラルはアプリケーションが必要とする場合のみ、オンにします。必要としない場合、低消費電力の状態またはオフにします。
- **未使用の GPIO:** 未使用の GPIO は一定のレベル (内部プルアップを持つ入力) に設定する必要があり、電力消費量を削減するために、入力を開放にしないことが必要です。
- **ホスト通信:** いくつかのシステムでは、ホストはシリアル プロトコル (I^2C、SPI や UART) を使って CapSense コントローラーと通信します。一般的な CapSense アプリケーションにおいて、センサー スキャンが完了した後、CapSense デバイスはスリープ モードに移行します。ホスト プロセッサとトランザクション (I^2C/SPI/UART で) があれば、デバイスはスリープ モードに移行せず、また、ディープスリープ モードにある場合このモードを終了します。そのため、頻度の高いトランザクションは平均の消費電力を増加させます。平均の消費電力を減少するために、専用のホスト割込みピンを実装し、CapSense コントローラーがホスト割込みパルスをホストに送信する時のみに、トランザクションを開始します。

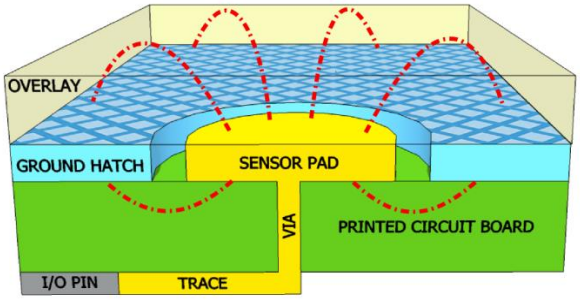
3 低消費電力向けのハードウェア設計上の注意事項

CapSense アプリケーションにおいて、静電容量センサーは、プリント基板 (PCB) またはフレックス回路の配線により形成されます。良好なハードウェア設計は、設計が堅牢で低い平均電力消費量の達成を確実にします。不十分に設計された CapSense ハードウェアは電力消費量を増加させ、ファームウェア アルゴリズムまたはセンサー チューニングで完全に補完できません。以下では低い平均電力消費量を達成するためのハードウェア設計上の注意事項を説明します。

3.1 寄生容量

電力消費量を低減するために、センサーの寄生容量を削減する必要があります。CapSense 自己容量システム (CSD) では、コントローラーが測定するセンサー静電容量は「Cs」と呼ばれます。センサーに指が乗っていなければ、Cs はシステムの寄生容量 (C_P) と等しくなります。図 5 に示すように、この寄生容量は、センサー パッド、オーバーレイ、CapSense コントローラー ピンとセンサー パッド間の配線、回路基板を貫通するビア、および CapSense コントローラーのピン静電容量の影響を含む、分布容量を単純化したものです。したがって、C_P の主な要素は、配線容量とセンサーの静電容量です。C_P は、センサー パッド周囲の電界に関連しています。高い C_P 値は電力消費量を増加させ、感度を低下させます。

図 5. C_P および電界



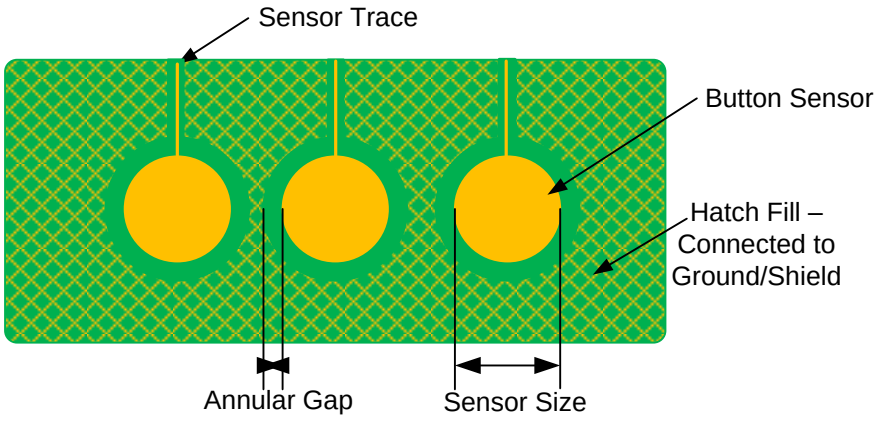
指がセンサー表面に触れている時は、オーバーレイを通して、センサー パッドと指の間の簡単な平行板コンデンサを形成します。その結果の静電容量は「指の静電容量、Cf」と呼ばれます。Cf は、人体および回路グランドからの戻り経路の影響を含む分布容量の単純化です。

静電容量センス コントローラーは CSD 法を使用して、その入力ピンで測定される静電容量をデジタル カウントに変換します。コントローラーはデジタル カウントで指の接触によるセンサー静電容量の増加を識別します。

指の接触を正確に検出するために、CSD のスキャン分解能パラメーターを調整して、特定の感度レベルを維持する必要があります。C_P 値が高くなる場合、信頼性の高い感度を達成するために、センサーのスキャン分解能を増やす必要があります。式 3 に示すように、分解能が増加すると、変換時間 (スキャン時間) は長くなります。結果として、静電容量センス アプリケーションの平均の電力消費量は高くなります。電力消費量を削減するために、センサーC_p 値を減らして、より低いスキャン分解能をセットします。

- **センサー サイズ:** センサーは小さければ小さいほど、C_P 値は低くなります (図 6)。C_P が低くなると、必要な信号量を達成するために必要なスキャン分解能は低くなり、それによって平均の消費電力量が低下します。しかし、センサー サイズがオーバーレイの厚さおよび感度パラメーターに対して小さすぎる場合、指タッチを検出できないことがあります。オーバーレイの厚さの要件および感度パラメーターの設定に応じて、最適なセンサー サイズを使用してください。AN85951 – PSoC 4 CapSense Design Guide を参照してください。

図 6. CapSense センサー レイアウト



- **配線:** 配線の長さと幅はセンサーC_P に追加されるため、最小にする必要があります。チップをセンサーに近く配置して配線の長さを最小にし、それによって C_P 値を減らせます。
- **環状ギャップ:** センサーとグランドの間のギャップが大きくなると、ノイズ耐性が低下します。したがって、C_P 値を最適にするために、(システムの要件に応じて) 消費電力量およびノイズ耐性の間にはトレードオフ関係が生じます。センサーCP 値および感度への環状ギャップの影響についての詳細については Getting Started with CapSense を参照してください。
- **ハッチ グランド フィル:**ベタ グランド面の代わりに、ハッチンググランドプレーンを使用して C_P 値を減らします。センサー周囲およびセンサー下で PCB の最下層上では、ハッチング グランド面を使用することを推奨します。
- **シールド**: センサーの C_P を低減するためには被駆動シールド信号で最上層と最下層でのハッチフィルを駆動する必要があります。最上層で 0.17mm (7mil) 配線と 1.143mm (45mil) グリッドのハッチフィルを、最下層で 0.17mm (7mil) 配線と 1.778mm (70mil) グリッドのハッチフィルを使用し、このハッチを被駆動シールド信号で駆動します。
- ループ センサーを使用することで、近接センサーの C_P を最小限にできます。
- FPC では、C_P を最小にするために、ハッチフィルを最上層のみに使用します。ハッチ フィルはグランドまたは被駆動シールド信号のいずれかに接続できます。

3.2 オーバーレイ

オーバーレイは CapSense センサーのタッチ面として機能する非導電性素材です。オーバーレイの素材と厚さは指の静電容量 (C_F) を決定します。式 4 に示すように、指の静電容量は オーバーレイ素材の比誘電率に正比例し、オーバーレイの厚さに反比例します。指の静電容量が低くなると、信号も低くなります。これを補償するために分解能を高める必要がありますが、それによってスキャン時間と電力消費量が増加します。

$$C_F = \frac{\varepsilon_0 \varepsilon_r A}{d} \qquad \text{式 3}$$

したがって、低消費電力のためにはオーバーレイの厚さを小さくし、比誘電率の高いオーバーレイ素材を使用します。一般的なオーバーレイ素材の比誘電率は表 1 のとおりです。2.0～8.0 の比誘電率の素材は、静電容量センシング アプリケーションに最適です。

表 1. 一般的素材の比誘電率

素材	εr
空気	1.0
フォーマイカ®	4.6～4.9
ガラス (一般的なもの)	7.6～8.0
ガラス (セラミック)	6.0
PET フィルム (Mylar®)	3.2
ポリカーボネート (Lexan®)	2.9～3.0
アクリル (Plexiglass®)	2.8
ABS	2.4～4.1
木造のテーブルおよびデスク	1.2～2.5
石膏（乾式壁)	2.5～6.0

4 ファームウェア設計上の注意事項

ここでは消費電力量を低下させ、なおかつ信頼性の高い動作を確保するファームウェア技術について説明します。主な目的は、デバイスがほとんどの時間で低消費電力モードになるように、タスクを最適化するものです。PSoC 4 の低消費電力モードでは、本質的な機能性を保持しながら、全体の電力消費を低減できます。

4.1 マニュアル調整対自動調整

CapSense CSD 方式はハードウェアとファームウェア技術を合わせたものです。したがって、適切な動作のために必要ないくつかのハードウェアとファームウェアのパラメーターがあります。これらのパラメーターは最適な性能を達成するために調整する必要があります。市場での静電容量タッチソリューションのほとんどは手動で調整しなければなりません。サイプレスは、SmartSense (自動チューニングとしても知られる) と呼ばれる PSoC 4 CapSense のユニークな機能を提供します。

SmartSense は、すべてのパラメーターを最適な値に自動的に設定するファームウェアアルゴリズムです。SmartSense 自動チューニングは開発期間を短縮し、PCB のばらつきに対して安定した性能を提供します。しかし、実行時に CapSense パラメーターのチューニングを行えるように、追加 RAM と CPU リソースを必要とします。

手動チューニングは最適な CapSense パラメーターをチューニングするためにある程度の努力が必要ですが、それにより応答時間と電力消費量などの静電容量センス システムの特性を厳密に制御できます。手動チューニングにより、センサーのスキャン時間を厳密に制御でき、結果としてアプリケーション エンジニアはできるだけ低い電力消費量を得るように調整できます。

各調整モードで電力消費量は以下の順で増えます。

手動チューニング ≤ SmartSense (ハードウェア パラメーター) < SmartSense (フル自動チューニング)

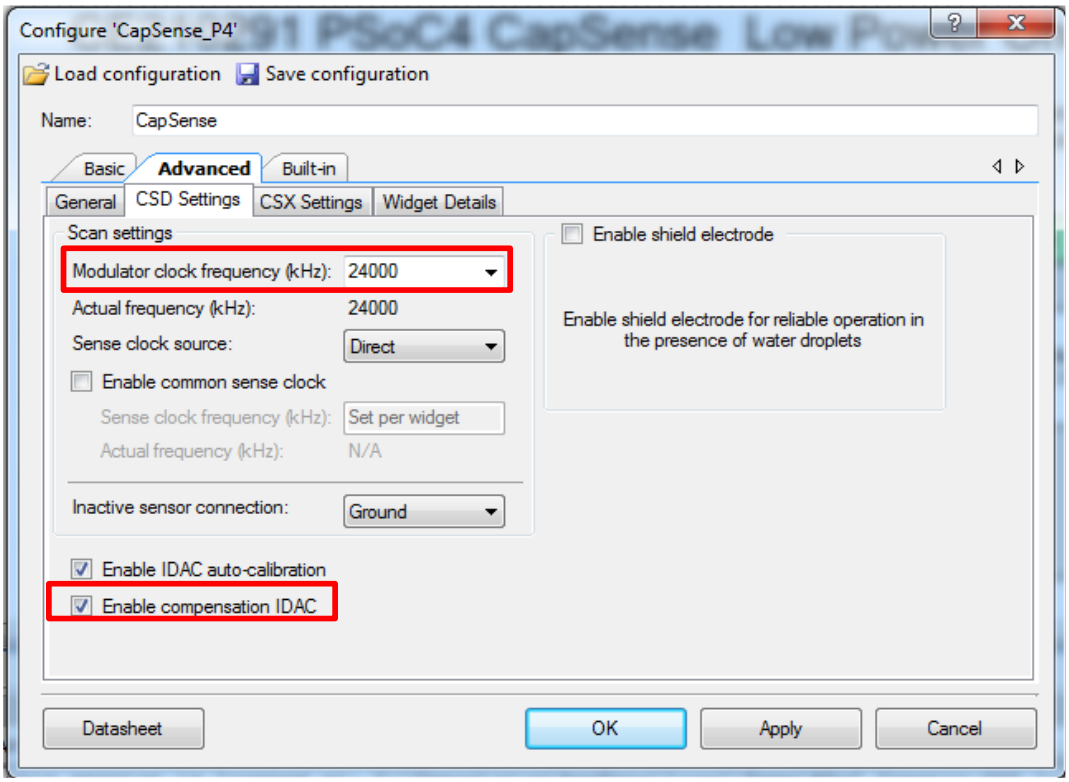
4.2 低消費電力のための CapSense コンポーネントの設定

設計者は CapSense コンポーネントのコンフィギュレーション パラメーターを調整して平均の消費電力量を削減します。コンポーネント パラメーターを最適な値に設定することで、スキャン時間を減らせます。それによりアクティブ モードでの時間が短くなり、スリープ時間が長くなり、消費電力量が低くなります。CapSense コンポーネントで、補正 IDAC、変調器周波数、およびスキャン分解能の 3 つの主なパラメーターを平均の消費電力量を低くするように設定できます。

補正 IDAC の有効化: 補正 IDAC を有効にすると、システムがより低い分解能を使用して同じ信号量を達成できるため、消費電力が低減します。分解能を低くすると、スキャン時間が短くなり、電力消費量が低くなります。図 7 に示すように、補正 IDAC は Advanced Tab/CSD Settings で有効にします。補正 IDAC は汎用 DAC 要件に必要とされない限り、センサー寄生容量が約 20pF より高い場合、有効にすることを推奨します。

変調器周波数: 変調器周波数は高ければ高いほど、スキャン時間が短くなり、そのため消費電力量が低くなります。図 7 に示すように、このパラメーターは Advanced Tab/CSD Settings にあります。

図 7. CapSense コンポーネントの Advanced Tab/CSD Settings



スキャン分解能／感度: スキャン分解能は raw カウント カウンターの分解能です。式 3 に示すように、センサーのスキャン時間はスキャン分解能パラメーターに正比例します。分解能が 1 ビット増えるたびに スキャン時間が倍になります。

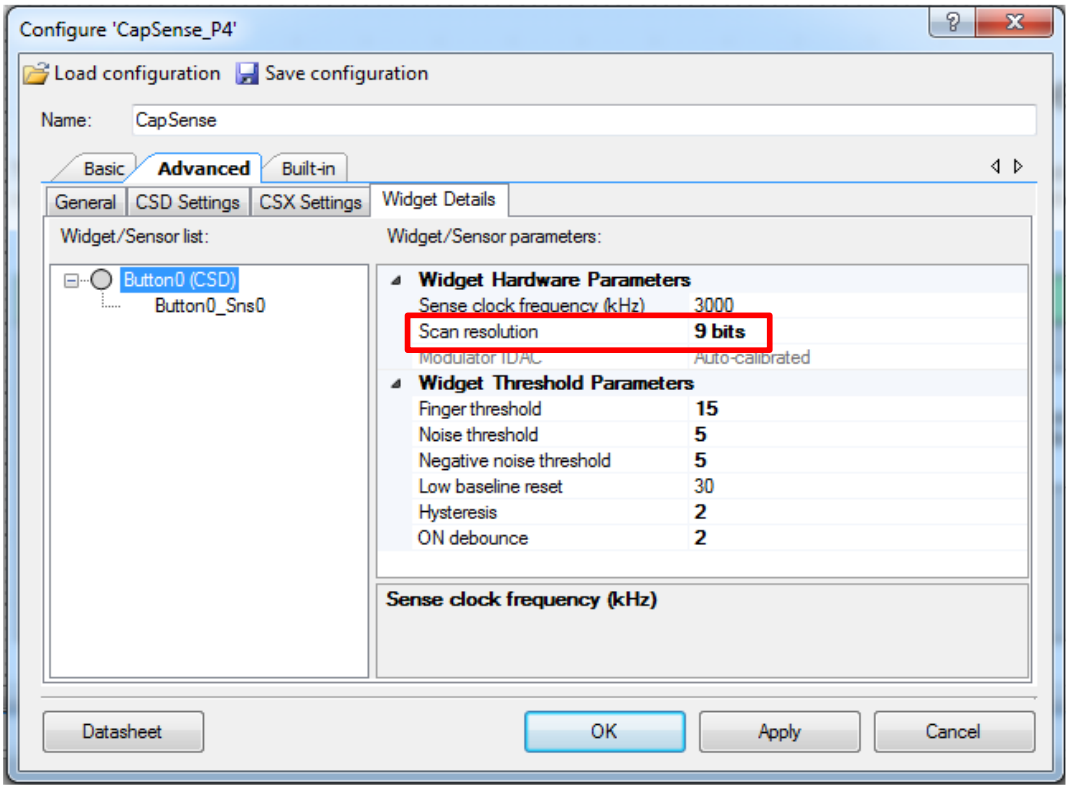
$$Sensor\ scan\ time = \left(2^{Resolution} - 1\right) \Big/ Modulator\ Clock\ Frequency \qquad \text{式 4}$$

ここで、変調器クロック周波数は raw カウント カウンターにより使用されるクロック周波数です。

感度パラメーターはスキャン分解能パラメーターと同様であり、指の静電容量 (C_F) 1pF あたりの信号カウントとして定義されます。チューニング モードが SmartSense である場合、このパラメーターはスキャン分解能の代わりに使用されます。この感度パラメーターは、センサー信号の強度を増減させます。図 3 に示すように、感度パラメーターはセンサーのスキャン分解能およびスキャン時間に直接影響を与えます。すなわち、感度パラメーターの値を高めると、スキャン分解能が高くなり、スキャン時間が長くなります。その結果として、感度のパラメーター値が高いほど、消費電力量は増加します。

手動チューニングを使用する時、分解能が低いほどスキャン時間は短くなり、そのため消費電力量も低くなります。しかし、低い分解能はタッチ応答および SNR に影響を与えます。したがって、これは設計エンジニアが考慮すべきトレードオフです。図 7 に示すように、このパラメーターは Advanced/Widget Details 設定にあります。SmartSense を使用する場合、感度パラメーターはセンサーのスキャン時間および消費電力量を決定します。感度設定が低くなると消費電力量も低くなります。このパラメーターは、SNR が 5:1 比より高くなるように設定する必要があります。

図 8. CapSense コンポーネントの Advanced タブ／Widget Details 設定



4.3 電力モードの動的切り替え

デバイス電力モードの動的切り替えにより、デバイスの平均電力消費量を減らせます。

4.3.1 スキャン間でディープスリープ モードを使用

CapSense 設計の電力消費量を大幅に削減するために、PSoC 4 のディープスリープモードを使用するべきです。CapSense のスキャンとスキャンの間でシステムが最小消費電力モード (ディープスリープ モード) であることを確保することにより、電力消費量を削減できます。

ディープスリープ モードでは CapSense のハードウェアは無効です。そのため、デバイスは頻繁にウェイクアップして、タッチをスキャンし、システムが必要とする動作を実行する必要があります。PSoC 4 での WDT を使用して、デバイスを頻繁な時間間隔でディープスリープ モードからウェイクアップできます。

一般的なアプリケーションでは、ディープスリープ モードからウェイクアップするために以下の方法があります。

WDT を使った定期的なスキャン

WDT を使用して、PSoC 4 デバイスを一定の時間ディープスリープ モードに移行させ、CapSense センサーのスキャンのために時々ウェイクアップできます。一般的には人間とユーザー インターフェース (UI) の相互作用、ユーザーがディープスリープ中に CapSense の非アクティブ状態を認識しないため、この手法は有用です。

CapSense アプリケーションでは、システムをウェイクアップさせる周波数はユーザーの応答時間によって決められます。通常、1 回の有効な指のタッチは、数十ミリ秒から数百ミリ秒の間継続します。そのようなシステムの場合、PSoC 4 はミリ秒ごとにウェイクアップして指のタッチをスキャンする必要はありません。ウェイクアップ間隔は CapSense スキャン レートを決定し、アプリケーションによって異なります。

GPIO 割込み

PSoC 4 は GPIO へのデジタル入力信号の立ち上りエッジ、立ち下りエッジ、または両方のエッジで生成される割込みによって低消費電力モードからウェイクアップできます。GPIO 割込み、ホスト プロセッサ、変換器、またはユーザー ボタンを使って、GPIO でトリガーを行うことにより PSoC デバイスをウェイクアップできます。GPIO ベースのウェイクアップにより、ディープスリープ モードでの消費電流は可能な限り低くなります。

I^2C アドレス一致

PSoC 4 デバイスのシリアル通信ブロック (SCB) ではマスターまたはスレーブとして動作する I^2C 機能があります。I^2C がスレーブとして設定されると、それによって PSoC デバイスをディープスリープ モードからウェイクアップできます。この方法により、リモート マスターは I^2C アドレス一致を介してターゲット PSoC デバイスをウェイクアップすることが可能です。その時、PSoC デバイスはアクティブ電力モードでの必要な処理を完了し、I^2C マスターに結果を送信し、ディープスリープ モードに戻ります。

I^2C アドレス一致により、ホスト コントローラーの電力モードや動作モードに関わらず、デバイスはどの電力モードにもなれます。ホストとスレーブ (PSoC 4) 間の同期化は必要ではありません。スレーブは必要な場合にのみウェイクアップし、動作を完了してディープスリープに戻ります。また、これにより、ホストが I^2C バスで他のデバイスと通信している時、望ましくないウェイクアップ イベントも回避できます。

これらの割込みソースを設定する方法の詳細については、AN90799 – PSoC 4 Interrupts を参照してください。

4.3.2 CapSense スキャン中にスリープ モードへの移行および維持

CapSense スキャンは非ブロッキングです。CapSense スキャンの起動と終了の間で CPU の介入は必要ありません。スキャン開始後、デバイスをスリープ モードに移行させて節電します。CapSense ハードウェアはスキャンを完了した後、割込みを生成して、デバイスをアクティブ電力モードに復帰させます。図 9 と図 10 に、これらの概念を示します。

図 9. タッチがない時のタイミング図

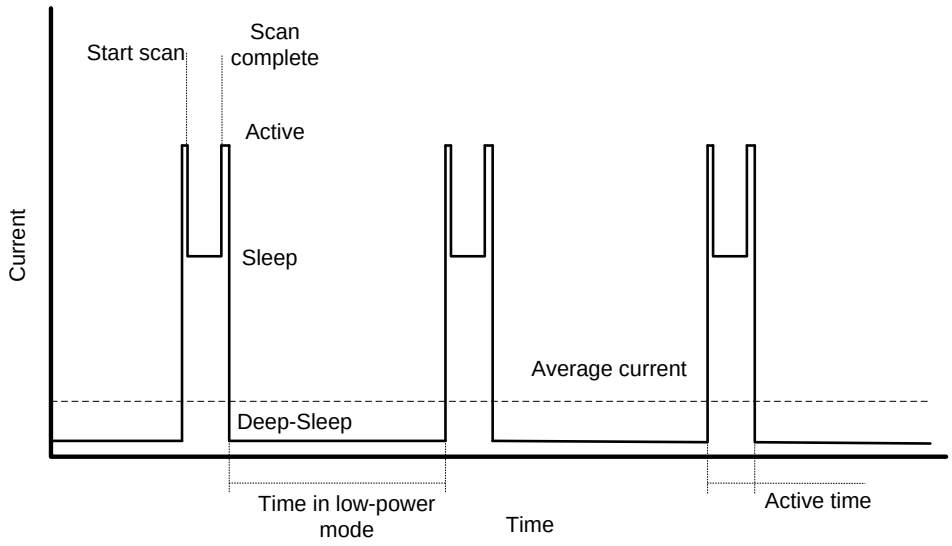
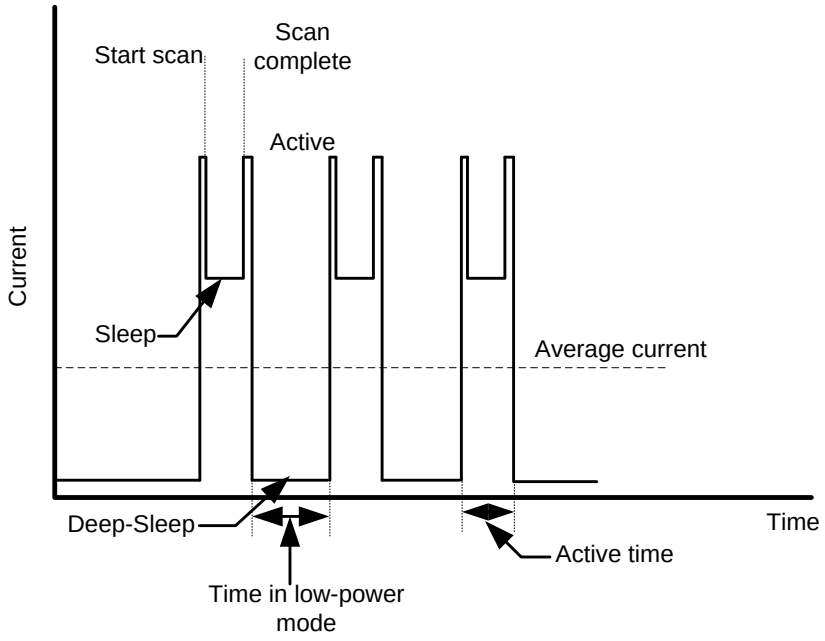


図 10. タッチがある時のタイミング図



4.4 リフレッシュ レート

アクティブ時間に影響する要素で説明したように、リフレッシュ レートは消費電力および性能を最適化するためにファームウェアで設定可能なパラメーターです。サンプル プロジェクトでは、応答時間の最適化および電力消費量の削減の方法について説明します。

5 低消費電力のためのシステム レベルの考慮事項

5.1 連結化センサーのスキャン

連結化センサー サンプリングは、平均消費電力を削減するために使用できます。この方法では、システムのスタンドバイ モードの間にシステムをウェイクアップできるすべての物理的センサーは、設計で単一の仮想連結化センサーを形成するように一緒に接続されます。連結化センサーのみをスキャンすることは、すべてのセンサーをスキャンすることより短時間ですむため、静電容量コントローラーはより長時間スリープ モードに維持できます。したがって、平均消費電力を削減します。

いずれかの物理的センサーがタッチされると、連結化センサーのセンサー静電容量が増加し、タッチが検出されます。しかし、連結化センサーのタッチを検知している時、スタンドバイ モードの間にタッチされた特定のボタンを確定できません。

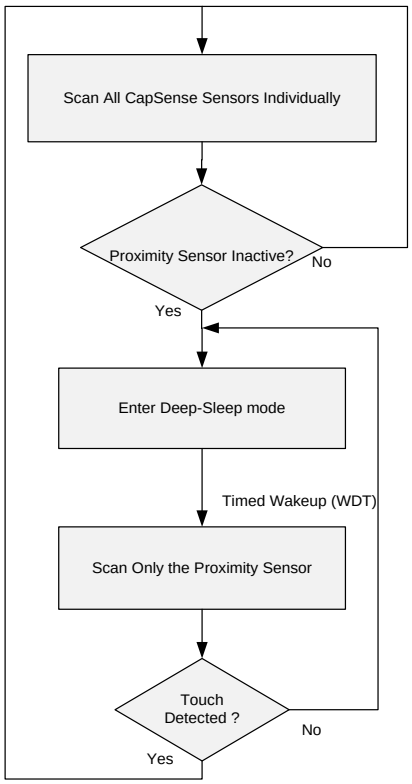
スタンドバイ モードの間にタッチされた特定のボタンを確定するために、静電容量コントローラーはアクティブ モードに復帰する必要があります。物理的センサーは、連結化センサーとの接続を切断し、タッチされたセンサーを検出するために個別にスキャンする必要があります。

この方法では、複数の物理的センサーを組み合わせて単一の仮想連結化センサー形成することによって平均消費電力を最適化し、静電容量コントローラーがタッチを検出した場合にのみアクティブ モードに戻ることを確保します。静電容量コントローラーがアクティブ モードに移行した後、指のタッチが一定の期間 UI パネルで検出されない場合、静電容量コントローラーは連結化センサー モードに戻す必要があります。「CE210290 - PSoC4 CapSense Low Power Ganged Sensor」のサンプル コードは平均電力消費量の削減のために連結化スライダーを実装する方法について示します。

5.2 近接センサーを使用したウェイク オン アプローチ

設計でのセンサーの数が増えると、すべてのセンサーをスキャンするためにデバイスはアクティブ モードでより多くの時間を費やします。したがって、平均消費電力を増加させます。設計で多くのセンサーを使用する場合、センサーを取り囲む別の近接センサー ループをおくべきです。この方法では、システムをウェイクアップするために物理的タッチを必要としません。デバイスがディープスリープ モードから復帰した時、近接センサーのみをスキャンして総スキャン時間を減らし、これにより平均電力消費量を削減します。近接センサーがアクティブである (ユーザーの手が UI パネルに近づく) 場合、デバイスはアクティブ モードのままで他のセンサーをスキャンする必要があります。近接センサーが非アクティブであれば、デバイスはディープスリープ モードに戻せます。図 11 に、このプロセスを示します。

図 11. 近接センサーを使用したウェイク オン アプローチ

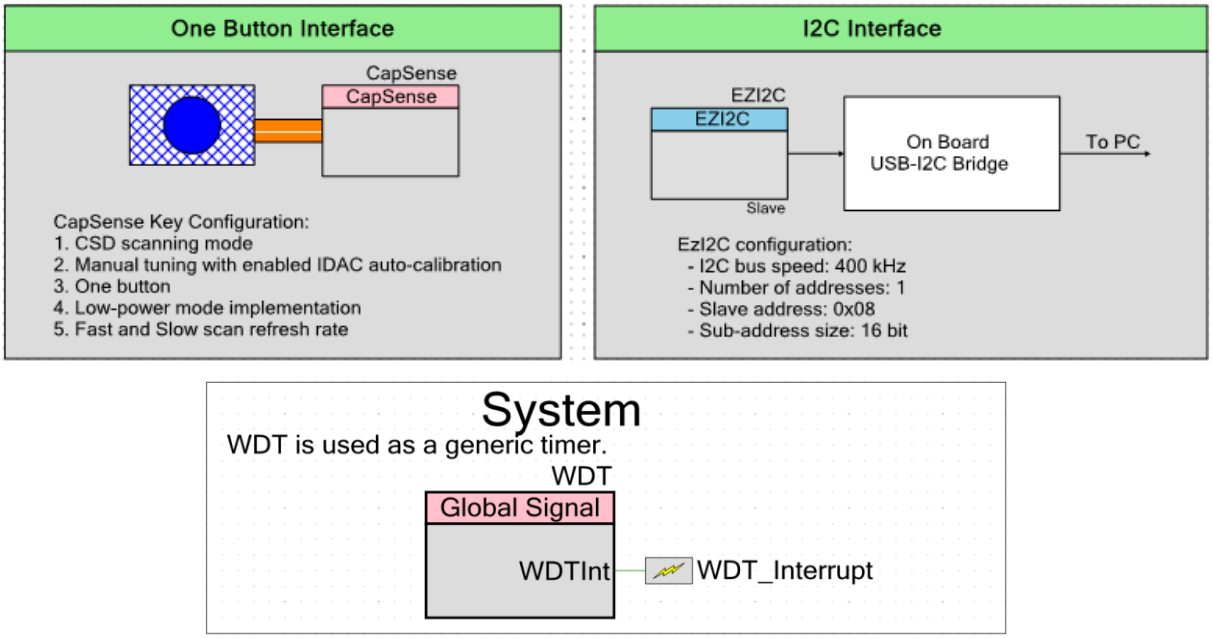


さらに、消費電力を削減するために、近接センサーを使用して UI パネルのバックライトを制御します。静電容量コントローラーがスタンドバイ モードである (アクティブなセンサーがない) 場合、バックライトをオフにして装置の非アクティブ状態を示します。ユーザーの手がパネルに近づき、近接センサーがそれを検出すると、バックライトはオンになって、ユーザーに正しいボタンをタッチすることを支援します。これにより、エンド システムでの全体的な消費電力量も削減できます。

6 サンプル プロジェクト

このサンプル コード「CE210291 – PSoC4 CapSense® Low Power One Button」は、応答時間に影響を与えず CapSense システムの平均電力消費量を減らす方法について紹介します。CapSense コンポーネントはワンボタン ウィジェットで構成されます。このプロジェクトは IDAC 自動較正が有効になる CSD 手動チューニング モードを使用します。EZI2C コンポーネントは I^2C を介して外部ホスト／チューナにセンサー データおよびボタンのタッチ位置情報を送信します。グローバル信号リファレンス コンポーネントは、リフレッシュ レートを設定するために使用されるウォッチ ドッグ タイマー (WDT) として設定されます。

図 12. コンポーネントの回路図



このサンプル コードでは、アクティブ モードで 45[1]Hz および Look-for-Touch (LFT) モードで 8Hz という 2 つのリフレッシュ レートを実装します。これらのリフレッシュ レートは、許容電力に基づいて、最適な応答時間を達成するために選択されます。図 13 に示すように、最初、デバイスは 8Hz でセンサーをスキャンする LFT モードにあります。タッチが検出されると、デバイスは図 14 に示すように、速い指のタッチを検出するために 45Hz でセンサーをスキャンするアクティブ モードに切り替わります。特定期間にタッチが検出されない場合、デバイスは平均消費電力を下げるためにリフレッシュ レートを 8Hz に変更します。平均消費電力はアクティブ モードで約 20µA で、Look-For-Touch モードで約 6µA です。

[1] 45Hz のリフレッシュ レートはアクティブ モードで 20µA の平均消費電流に対応しながら、最適なボタン応答を確保するために選択されます。

図 13. アクティブスキャン モードのタイミング図

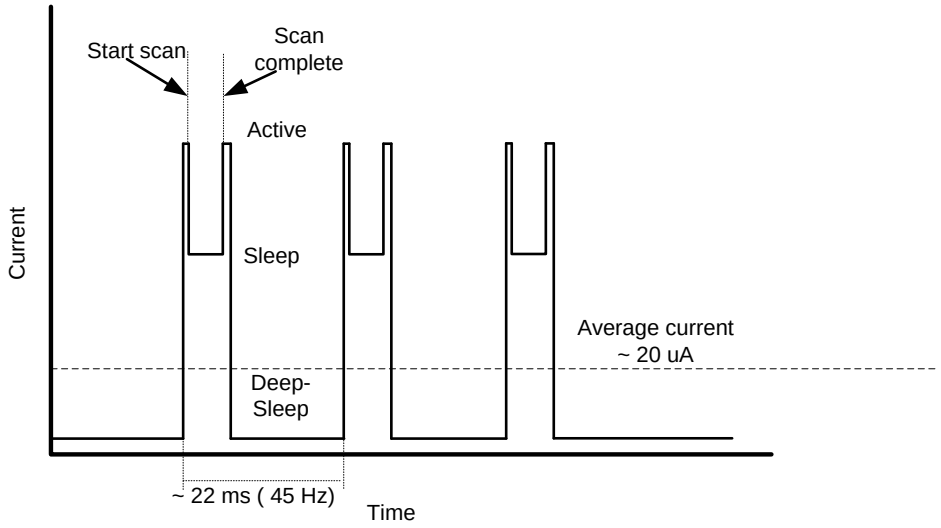
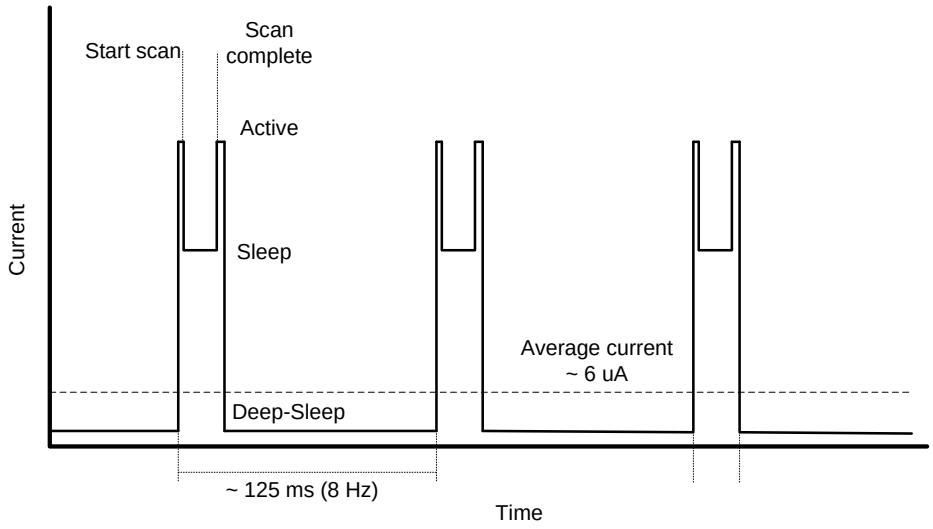


図 14. LFT モードのタイミング図



このサンプル コードは PSoC 4000S および関連する CY8CKIT-041-40xx キット向けに設計されています。一般的には、デバイスおよびコンポーネント用のピンの割り当てを変更するだけで、デザインを他の PSoC 4 デバイスおよびキットに簡単に移植できます。ファームウェア実装の詳細については CE210291 資料を参照してください。

7 平均電流測定

サンプル プロジェクトを使用して、平均電流消費量を測定し、電流プロファイルを観測できます。デバイスが消費する瞬間電流は一定値でなく、電力モード遷移によって動的に変化するチップの状態によって変わります。したがって、これらの電流バーストの期間は非常に短いため、ハンドヘルド マルチメーターを使って個々の瞬間電流を計測することは困難です。したがって、測定の「開口」を設定するオプションのあるマルチメーターを使用する必要があります。開口は期間「T」であり、この期間中にマルチメーターが瞬間電流を測定し、統合し、その後、期間「T」の平均電流を表示します。関連するキット ユーザー ガイドおよびサンプル コードの資料は、デバイス消費電流の測定方法について詳細に説明します。測定手順を以下にまとめます。

1. サンプル プロジェクトを構築し、キットに hex ファイルをプログラムしてください。

2. Agilent 34411A 6 1/2 のデジタル マルチメーターなどの 1 つのマルチメーターを使って、マルチメーターをキットの電流測定ジャンパに接続してください。
3. スキャン期間に基づいて測定用の開口を設定してください。正確な開口を設定できない場合、スキャン期間の整数倍に設定します。
4. 電流を測定してください。測定される電流は 1 間隔での平均電流です。

8 まとめ

AN210998 は低消費電力 CapSense システム設計に関するハードウェアとファームウェアの注意事項について紹介します。それに関連するサンプル プロジェクトは、CapSense ベースのアプリケーションで優れたタッチ応答を維持しながら超低消費電力を達成する方法について説明します。

電力消費量は、良いアイデアと実際の成功したデザインとの違いとなり得ます。PSoC 4 で使用可能な各省電力機能をうまく利用することで、設計を最適化し、最低の電力消費量を確保できます。

9 関連リソース

これらのアプリケーション ノートおよびサンプル コードは、本書の範囲内では十分説明しきれていないトピックに関する詳しい情報を提供します。

- AN79953 - PSoC 4 入門
- AN85951 – PSoC® 4 and PSoC® 6 MCU CapSense® Design Guide
- AN64846 - Getting Started with CapSense®
- AN86233 - PSoC 4 の低消費電力モードおよび消費電力低減技術
- AN77900 - PSoC® 3 と PSoC 5LP の低電力モードおよび電力低減技術
- CE95288 - CapSense Low Power with PSoC 4
- CE210291 - PSoC4 CapSense Low Power One Button
- CE210290 - PSoC4 CapSense Low Power Ganged Sensor
- CE195286 – PSoC® 4 CapSense Tuner

著者について

名前: Vijay Kumar Marrivagu

役職: 主任システム エンジニア

経歴: デジタル設計と検証で数年の経験があります。

A 付録 A. 電力モードのまとめ

ここでは、PSoC 4 デバイスで利用可能な電力モードを詳しく説明します。PSoC 4 はアクティブ、スリープおよびディープスリープの 3 つの動作モードをサポートします。これらの電力モードの差異は消費電力とペリフェラル利用可能性です。

A.1 アクティブ

アクティブ モードは、デバイスの主要な動作モードであり、起動時のデフォルト モードです。このモードで、内部主発振器 (IMO) から生成された高周波数内部クロック (HFCLK) はオンであり、CPU とすべてのペリフェラルはファームウェアによって特に無効化されない限り動作可能です。一般的に、アクティブ モードでチップのすべてのブロックが動作状態になるため、このモードでの消費電力は他のモードより大きくなります。

アクティブ モードは、他のどのモードにも移行できます。他のモードからの有効な割込みやリセット イベントは、PSoC 4 デバイスをアクティブ モードに戻します。

A.2 スリープ

スリープ モードは、アクティブ モードと同じように、デバイス内のすべてのペリフェラルが動作状態になる低消費電力モードです。スリープ モードでは、CPU のクロック (システム クロック (SYSCLK)) はオフです。CPU は割込みによりスリープ モードからウェイクアップできます。このモードで、CPU はデバイスの消費電力を減らすためにオフですが、CapSense や I^2C インターフェースなどのすべてのペリフェラルは動作可能です。

A.3 ディープスリープ

ディープスリープは PSoC 4 デバイスの最低消費電力モードです。このモードで、CPU、IMO、CapSense や TCPWM 等、ほとんどのペリフェラルはオフまたは保持状態になります。

しかし、このモードで I^2C、WDT および GPIO など基本ペリフェラルは動作状態になります。I^2C ブロックはコンフィギュレーションに従ってアドレス一致を介してデバイスをディープスリープ モードからウェイクアップできます。ILO クロック ソースもディープスリープ モードで動作可能になります。クロック ブロックで実装されている WDT により、ディープスリープからの定期的なウェイクアップは可能となります。

電力消費量の詳細については、デバイスのデータシートを参照してください。

A.4 モードへの移行およびウェイクアップ ソース

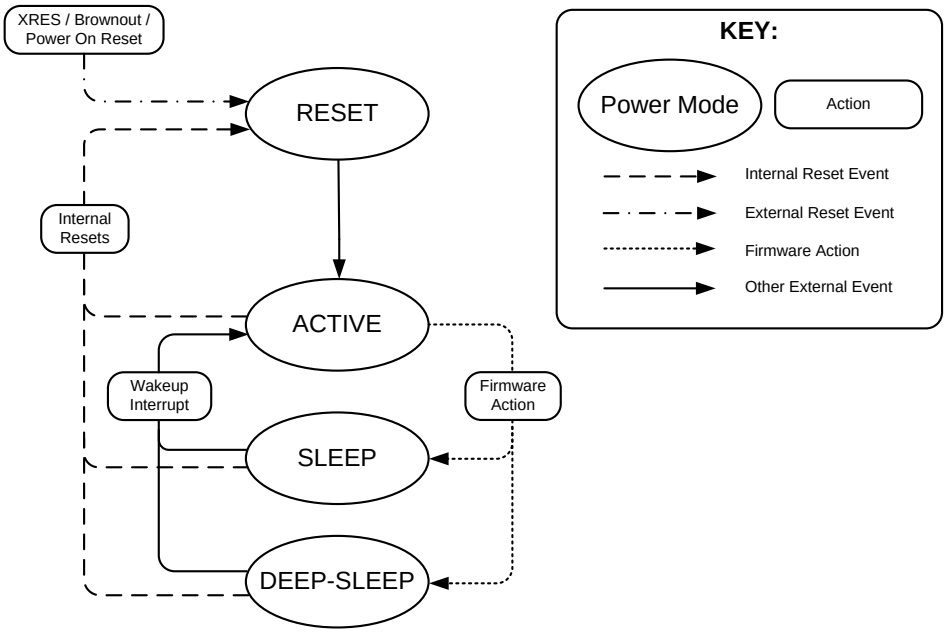
PSoC Creator は、低消費電力モードに移行するためのレジスタ レベルの動作を抽象化する「アプリケーション プログラミング インターフェース (API)」と呼ばれる関数セットを提供します。表 2 は低消費電力モードに移行するための API を示します。これは、低消費電力モードを終了してアクティブ モードに戻すためのウェイクアップ ソースを識別します。図 12 に消費電力モード間の可能な遷移を示します。

これらの低消費電力モードおよびウェイクアップ ソースの詳細は PSoC 4 Architecture TRM を参照してください。PSoC 4 での割込みの詳細については、AN90799 – PSoC 4 Interrupts を参照してください。

表 2. モード遷移の詳細

低消費電力モード	低消費電力モードに移行するための API	ウェイクアップ ソース	ウェイクアップ動作
スリープ	CySysPmSleep()	任意の割込みソース	割込み
		任意のリセット ソース	リセット
ディープスリープ	CySysPmDeepSleep()	GPIO 割込み	割込み
		I^2C アドレス一致	割込み
		ウォッチドッグ タイマー	割込み／リセット
		XRES (外部リセット ピン)、ブラウンアウト	リセット

図 15. パワー モードの遷移



B 付録 B: 用語集

用語	定義
静電容量センサー	静電容量の変化によってタッチまたは近づいている物体に反応する導電体および基板 (PCB 上の銅ボタンなど)
CapSense	サイプレスのタッチ センス ユーザー インターフェース ソリューション
補正 IDAC	過剰なセンサー寄生容量を補正するために CSD により使用されるプログラム可能な定電流源。この IDAC は、変調 IDAC とは異なり、CSD ブロックでシグマ-デルタ変調器によって制御されません。
CSD (CapSense シグマ デルタ方式)	サイプレスが特許権を有する、静電容量センスのアプリケーション用に自己容量(self-cap とも呼ばれる)を測定する方法。CSD モードでは、センス システムは電極の自己容量を測定し、指の存在を識別するために自己容量の変化が検出されます。
被駆動シールド	耐液性を可能にするために CSD により使用される技術であり、ここで、シールド電極が、センサー スイッチング信号に等しい位相および振幅を持つ信号により駆動されます。
電極	PCB 上のパッドや層、IOT、FPCB などの導電材料。電極は CapSense デバイスのポート ピンに接続され、CapSense センサーとして使用されるか、または CapSense の機能に関連した特定の信号を駆動するために使用されます。
連結化センサー	複数のセンサーを接続して、単一のセンサーとしてスキャンする方法。近接センス用のセンサーの面積を増やし、電力消費量を削減するために用いられます。システムが低消費電力モードにある時に電力を削減するために、センサーを個別にスキャンする代わりに、すべてのセンサーを連結し、単一のセンサーとしてスキャンして時間を短縮します。ユーザーがセンサーをタッチすると、システムはアクティブ モードに遷移して、アクティブになったセンサーを検出するためにすべてのセンサーを個別にスキャンします。PSoC はファームウェアでセンサー連動をサポートします。すなわち、複数のセンサーを AMUXBUS に同時に接続してスキャンできます。
ハッチ フィル、ハッチ グランド、またはハッチド グランド	静電容量センスの PCB を設計する際に、ノイズ耐性を得るために、接地した銅面をセンサーの周囲に配置する必要があります。しかし、ベタ グランドを使用すれば、センサーの寄生容量が増加するため、望ましくありません。そのため、グランドを特別なハッチ パターンで埋める (ハッチ フィル) 必要があります。ハッチ パターンは、密接して置かれた交差してメッシュのように見えるラインを持っており、ラインの幅とラインの間隔が充填率を決定します。耐液性を得るために、このハッチ フィル (シールド電極と呼ばれる) はグランドの代わりにシールド信号で駆動されます。
IDAC (電流出力デジタル-アナログ変換器)	PSoC の内部にあり、CapSense および ADC の動作に使用されるプログラマブルな定電流源
リニア スライダー	指の物理的な位置 (単一の軸で) を検出するために特定の直線状で配置された複数のセンサーからなるウィジェット
手動チューニング	CapSense パラメーターを手動で設定する (または調整する) プロセス
変調器クロック	センサー スキャン中の CSD ブロックからの変調器の出力をサンプリングするために使用されるクロック ソース。このクロックは Raw カウントのカウンターにも供給されます。スキャン時間 (事前および事後処理時間を除く) は「(2N – 1)/変調器クロック周波数」の式で計算されます (ここで、N はスキャンの分解能) 。
変調 IDAC	変調 IDAC はプログラム可能な定電流源であり、その出力は CSD ブロック内のシグマ-デルタ変調器の出力によって制御 (オン／オフに) されて、AMUXBUS 電圧を VREF に維持します。この IDAC によって供給される平均電流はセンサー コンデンサが引き出した平均電流に等しくなります。
相互静電容量	ある電極 (例えば、TX) と他の電極 (例えば RX) 間の静電容量
ノイズ (CapSense ノイズ)	センサーがオフ状態にある (タッチなし) 時にピークツーピークのカウントとして測定される Raw カウント値の変化
オーバーレイ	静電容量センサーをカバーし、タッチ面として機能するプラスチックやガラスなどの非導電性材料。センサーを備えた PCB はオーバーレイの下に直接配置されるか、またはスプリングを介して接続される。製品の筐体は多くの場合にオーバーレイになります。
寄生容量 (CP)	寄生容量は PCB の配線、センサー パッド、ビアとエアギャップによるセンサー電極の固有容量です。寄生容量は CSD の感度を低下させるため、望ましくないものです。
近接センサー	あらゆる物理的な接触なしに近くの物体の存在を検知できるセンサー
ラジアル スライダー	指の物理的な位置を検出するために特定の円形状に配置された複数のセンサーからなるウィジェット

用語	定義
raw カウント	センサーの物理的静電容量を表す CapSense ハードウェア ブロックの未処理のデジタル カウントの出力
スキャン分解能	CSD ブロックによって生成される Raw カウントの分解能 (単位はビット)
スキャン時間	センサー スキャンの完了時間
リフレッシュ レート	リフレッシュ レートは CSD ブロックがセンサーをスキャンする周波数として定義され、指のタッチの検出時から報告時までの可能な最小時間を決めます。最大のリフレッシュ レートはセンサー スキャン時間によって制限されます。
自己容量	回路のグランドと電極間の静電容量
感度	センサー静電容量の変化に対応し、カウント／pF で表される Raw カウントの変化。センサーの感度は、基板レイアウト、オーバーレイ特性、センス方式およびチューニング パラメーターに依存します。
センス クロック	CSD センス方式のフロントエンド スイッチキャパシタを実装するためのクロック ソース
シールド電極	センサーの周囲を覆う銅トレースで水または他の液体による誤タッチを防止します。シールド電極は CSD ブロックからシールド信号出力によって駆動されます。被駆動シールドを参照してください。
信号対雑音比 (SNR)	タッチした時のセンサーの信号とタッチしない時のセンサーのノイズ信号との比率
SmartSense 自動チューニング	設計段階後に最適な性能を実現するためにパラメーターを自動的に設定し、システムや製造、環境の変化を連続的に補正する CapSense アルゴリズム
チューニング	CapSense の動作に必要な様々なハードウェアおよびソフトウェア又は閾値パラメーターの最適値を決定するプロセス
ウィジェット	単一センサーまたは同様のセンサー グループで構成される CapSense コンポーネントのユーザー インターフェース要素。ボタン、近接センサー、リニア スライダー、ラジアル スライダー、マトリックス ボタン、およびタッチパッドはサポートされているウィジェットです。

改訂履歴

文書名: AN210998 – PSoC 4 低消費電力 CapSense 設計

文書番号: 002-12473

版	ECN	発行日	変更内容
**	5255831	05/10/2016	これは英語版 002-10998 Rev. **を翻訳した日本語版 002-12473 Rev. **です。
*A	6913600	07/08/2020	これは英語版 002-10998 Rev. *B を翻訳した日本語版 002-12473 Rev. *A です。

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
An Infineon Technologies Company
198 Champion Court
San Jose, CA 95134-1709