CAPSENSE™による近接センシング

本書について

適用範囲と目的

本書は、インフィニオンの CAPSENSE™テクノロジーを使用した近接センシング ソリューションを設計および実装するための包括的な技術ガイドを提供します。

対象者

本書の対象読者は、主にインフィニオンの CAPSENSE™テクノロジーを使用した近接センシングソリューションの実装に関心のある開発者, エンジニア, および設計者です。

Note: このドキュメントは、インフィニオンの CAPSENSE™ テクノロジー, エレクトロニクス, 電気工学の原理, およびソフトウェア開発とプログラミングの経験についての基本的な理解を前提としています。インフィニオンが提供する CAPSENSE™ および PSoC™ アーキテクチャを初めて使用する場合は、参考資料セクションを参照してこれらの製品についてよく理解してください。

略語

Table 1　略語

略語	説明
ADC	Analog-to-Digital Converter (アナログ-デジタル変換器)
CMOD	Modulator Capacitor (変調器容量)
C_P	Parasitic Capacitance (寄生容量)
CSD	Self-Capacitance Sensing
CSH	Shield Tank Capacitor
CSX	Mutual-Capacitance Sensing (相互容量検出)
EMC	Electromagnetic Compatibility (電磁環境適合性)
ESD	Electrostatic Discharge (静電放電)
GND	Ground (グランド)
GPIO	General-Purpose Input/Output (汎用入出力)
HID	Human Interface Device
HMI	Human machine interface
IDAC	Current-Output Digital-to-Analog Converter (電流デジタル-アナログ変換器)
IDE	Integrated Development Environment (統合開発環境)
IIR	Infinite Impulse Response (filter)
MCU	Microcontroller Unit (マイクロコントローラー ユニット)
MSCLP	Multi Scan Converter version 3 – Low Power
PSoC™	Programmable System on Chip (インフィニオンのプログラマブル システムオンチップ コントローラー)
SNR	Signal-to-Noise Ratio (信号対ノイズ比)
VREF	CAPSENSE™ および ADC 動作に使用される PSoC™ 内で利用可能なプログラム可能な基準電圧ブロック

目次

本書について 1
略語 2
目次 3
1 はじめに 5
1.1 近接存在検知 5
1.2 ジェスチャ検知 5
1.3 アプリケーション 6
1.4 CAPSENSE™の概要 6
1.4.1 自己容量 7
1.4.2 CAPSENSE™システム 8
1.4.3 相互容量 9
2 CAPSENSE™ 近接センシング ソリューションの設計 10
2.1 用語集 11
2.2 課題 12
2.2.1 低電力 12
2.2.2 長い近接センシング距離 13
2.2.3 耐液性 14
2.2.4 高いセンシング信頼性/耐ノイズ性 14
2.2.5 方向性 15
2.3 CAPSENSE™ 近接センシングを開始するためのリソース 16
2.3.1 オンラインリソース 16
2.3.2 入門キット 16
2.3.3 サンプルコード 17
3 レイアウトガイドライン 18
3.1 存在検出 18
3.1.1 ボタンセンサー 18
3.1.2 連動センサー 19
3.1.3 PCB トレースセンサー 19
3.1.4 ワイヤーセンサー 20
3.2 一般的なレイアウトのガイドライン 20
3.2.1 センサーのサイズと近接センシング距離 20
3.2.2 センサーの寄生容量 20
3.2.3 シールド電極 21
3.3 外部コンデンサ 21
3.4 ピン配置 22
3.4.1 外部コンデンサのピン選択 23
3.5 導電性物体/金属が近くにある場合の検知 24
3.6 ジェスチャ検出のレイアウトガイドライン 25
3.7 耐液性に関するレイアウトのガイドライン 27
3.7.1 耐液近接センシング用のシールド電極の設計 27
3.7.2 耐液近接センシング用のガード センサーの設計 27
3.8 レイアウト シミュレーション 28
4 ファームウェア設計ガイドライン 29
4.1 ウィジェットの種類の設定 29
4.1.1 アクティブウィジェット 29
4.1.2 低電力ウィジェット 29
4.2 スキャン設定 30

目次

4.2.1 チャネルとスロットの設定 30
4.3 シールド設定 30
4.3.1 アクティブシールド 31
4.3.2 パッシブシールド 32
4.3.3 近接センシング用のシールドの設定 32
4.4 ウェイクオンアプローチ 33
4.5 センサーの連動 34
4.6 ジェスチャ検出ファームウェアのガイドライン 37
4.7 フィルター 39
4.7.1 ハードウェアフィルター 39
4.7.1.1 カスケード積分コム 2 (CIC2) フィルター 40
4.7.1.2 一次ハードウェア IIR フィルター 42
4.7.2 ソフトウェアフィルター 43
4.7.2.1 ソフトウェアの raw カウントフィルター 44
4.7.2.2 ソフトウェアベースラインフィルター 45
4.8 耐液性に関するファームウェアのガイドライン 45
4.8.1 自己容量の使用 45
4.8.2 CSX ガード センサーの使用 46
4.9 ミドルウェアライブラリと API 48
5 チューニングガイドライン 49
5.1 信号対雑音比 (SNR) 49
5.2 近接センシング設計の調整 51
5.3 耐液性のチューニング 56
5.3.1 CSD 技術による耐液性のチューニング 56
5.3.2 CSX ガードベースの耐液性技術のチューニング 58
5.4 内蔵セルフテスト (BIST) 59
5.4.1 ハードウェアテスト 59
5.4.2 ファームウェアテスト 59
5.5 トラブルシューティングのヒントとテクニック 60
参考資料 61
改訂履歴 62
免責事項 63

1 はじめに

近接センサーを使用すると、ユーザーは電子デバイスと対話できるようになり、デバイスは物理的に接触せずに近くの物体の存在を検出できるため、タッチレスコントロール, ジェスチャ コントロール, 物体検出, 近接スイッチなどのアプリケーションに最適です。

近接センシングは、容量センサー, 誘導センサー, 磁気センサー, ホール効果センサー, 光学センサー, 超音波センサー, およびレーダーなどのさまざまなテクノロジーを使用して実装できますが、それぞれに独自の長所と短所があります。

静電容量型近接センシングは、低コスト, 高信頼性, 低消費電力, 洗練された外観, 既存のユーザーインターフェースとのシームレスな統合を備えた堅牢な設計を可能にするため、広く採用されています。インフィニオンの CAPSENSE™デバイスは、30 cm の近接検知距離が可能な自己容量に基づく堅牢な近接検知機能を提供します。

このアプリケーションノートでは、次の方法について説明します。

- 静電容量型近接センシングの背後にあるテクノロジー
- インフィニオンの CAPSENSE™デバイスを使用した近接センサーの設計
- 堅牢で信頼性の高い近接センシング ソリューションのためのセンサーレイアウトガイドライン
- 近接検知ソリューションを実装するためにインフィニオンが提供するファームウェアライブラリとリソース
- 大きな近接検知距離と耐液性を実現するためのハードウェアおよびソフトウェアのパラメーターの調整ガイドライン
- 近接ベースのジェスチャ検出の実装に関するガイドライン
- トラブルシューティングのヒントとテクニック

1.1 近接存在検知

近接センシングは、物理的接触なしに近くの物体の存在を検出するプロセスです。CAPSENSE™に基づく静電容量式近接センシング技術は、通常数フェムトファラド (fF) の範囲にあるセンサーの静電容量の変化を測定することによって、(導電性) 物体の存在を検出します。 センサーは、ある長さのワイヤーまたは PCB 上の銅パッドである可能性があります。設計については、このドキュメントの後続のセクションで詳しく説明します。

1.2 ジェスチャ検知

ジェスチャ検出は、(人体の) 動きを解釈し、ジェスチャタイプの情報をデバイスに提供する技術です。ジェスチャベースのユーザー インターフェースは、システムと対話するための直感的な方法を提供します。これにより、コンピュータ画面上の世界地図の 3D 制御などのアプリケーションでユーザーエクスペリエンスが向上します。近接センサーは、非接触で左から右または右から左へのスワイプなどのジェスチャも検出できます。CAPSENSE™に基づく近接センサーを使用すると、ユーザーとデバイスの間の物理的な接触なしでジェスチャを検出できます。

1.3 アプリケーション

CAPSENSE™ は、一般的なヒューマンマシンインターフェーステクノロジーの 1 つであり、CAPSENSE™ ベースの静電容量型近接センシングは、次のようなさまざまなアプリケーションで広く使用されています。

- バッテリ駆動アプリケーションのウェイクオンアプローチ機能:
  - スマートロック
  - ワイヤレスマウス、コンピュータキーボード
- ヒューマン マシン インターフェース (HMI) でのジェスチャ検出:
  - 3D 制御コンピュータインターフェース
  - おもちゃ
  - 楽器 (テルミンなど)
- 存在と接近の検出:
  - ヒアラブル, スマートウォッチ, AR/VR メガネ, ヘッドセットなどのウェアラブル
  - タブレットおよび携帯電話における比吸収率 (SAR) 規制
- スマートホーム
  - スマート スピーカー, ドア ロック, 壁スイッチ, スマート ライト, 室内装飾, レンジフード, 調理台, コントロール パネルのバックライト制御
- 気候と利便性
  - 蛇口, ゴミ箱, トイレ, スマートコンポスト, サーモスタット (ジェスチャコントロール), 歯ブラシ (近づくとライトが点灯), 石鹸ディスペンサー, タオルディスペンサー, 自動ドアオープナーなど

1.4 CAPSENSE™の概要

インフィニオンの CAPSENSE™ コントローラーは、センサー表面上またはセンサー表面付近に指が存在することによる静電容量の変化が検出される静電容量センシングを使用します。通常、容量センサーは、PCB 上の円形の金属充填領域です。Figure 1 に、静電容量センサー ボタンの例を示します。センシング機能は、ハードウェアとファームウェアの組み合わせを使用して実現されます。

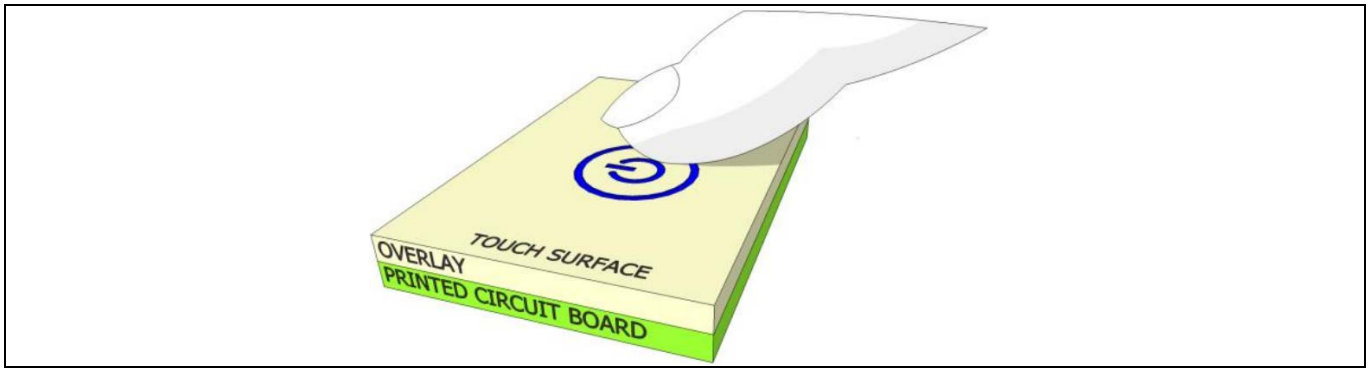


Figure 1 静電容量センサーの図

静電容量センサーは、Figure 2 に示すように、自己容量または相互容量という 2 つの異なる技術を使用して設計できます。

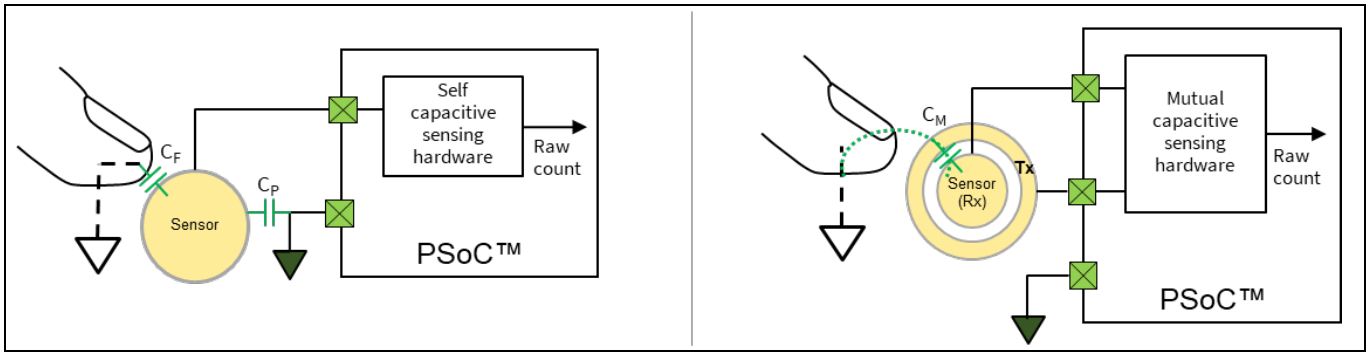


Figure 2　自己および相互静電容量ベースのタッチ センシングの動作原理

PSoC™ デバイスは、自己容量センシングには容量性シグマデルタ (CSD)、相互容量センシングには CAPSENSE™ クロスポイント (CSX) として知られるセンシング方法を使用します。CSD および CSX タッチ センシング方式は、業界最高の信号対雑音比 (SNR) を提供します。

1.4.1　自己容量

自己容量は単一のピンを使用し、ピンと近くのグランドの間の容量を測定します。CAPSENSE™自己容量システムでは、コントローラーによって測定されるセンサー容量は C_S と呼ばれます。

指がセンサーの近くにない場合:

センサー容量 (C_S) = システムの寄生容量 (C_P)

寄生容量 C_P 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 パッド間のトレース, 回路基板を通るビア, および CAPSENSE™コントローラーのピン容量の影響を含む分布容量を簡略化したものです。C_P はセンサーパッドの周囲の電場に関係します。Figure 3 にはセンサー パッドの周囲の力線のみが示されていますが、実際の電場はより複雑です。

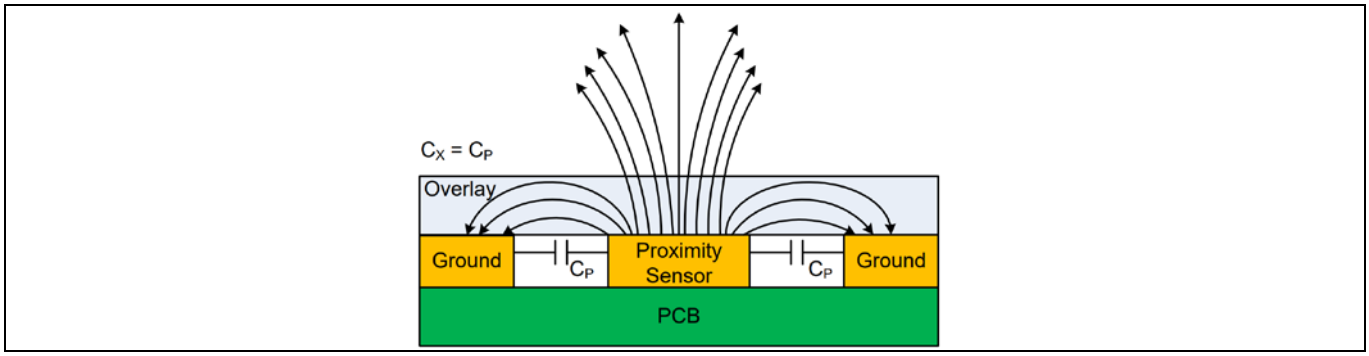


Figure 3　センサーの寄生容量と電界の図

指などの物体がセンサーに近づくと、Figure 4 に示すように、電力線の一部が対象物体に結合し、既存の C_P に少量の指の静電容量 (C_F) が追加されます。この静電容量の変化は次のようになります。CAPSENSE™回路によって測定され、対象物の近接を検出します。

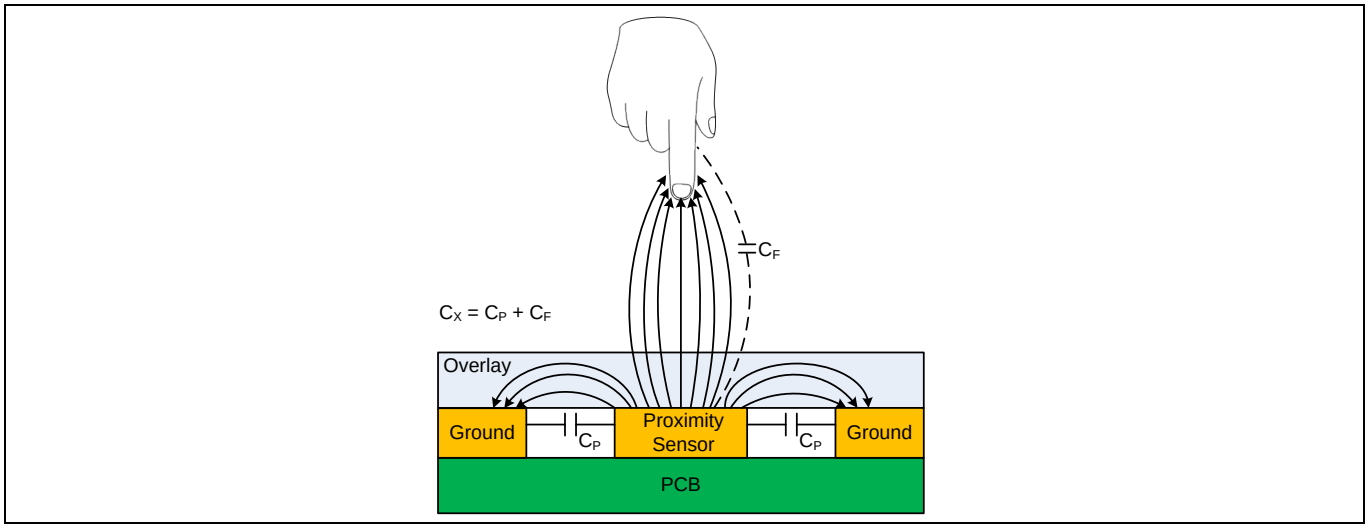


Figure 4　指に結合する電界線の図

指がセンサーに近づいた場合:

センサー容量 (C_S) = 寄生容量 (C_P) + 指容量 (C_F)

指の静電容量 (C_F) の追加による総センサー静電容量 (C_S) の変化が測定され、システムとのユーザー インタラクションが判断されます。

1.4.2　CAPSENSE™システム

インフィニオンの PSoC™デバイスは、Figure 5 に示すように、自己容量センシングに CAPSENSE™システムを使用します。このシステムは、PSoC™ピンに接続されたセンサー容量を充電し、容量を測定することによって動作します。次に、静電容量 - デジタルコンバーターが、それを「raw カウント」として知られる電極間の自己静電容量に比例するデジタル値に変換します。

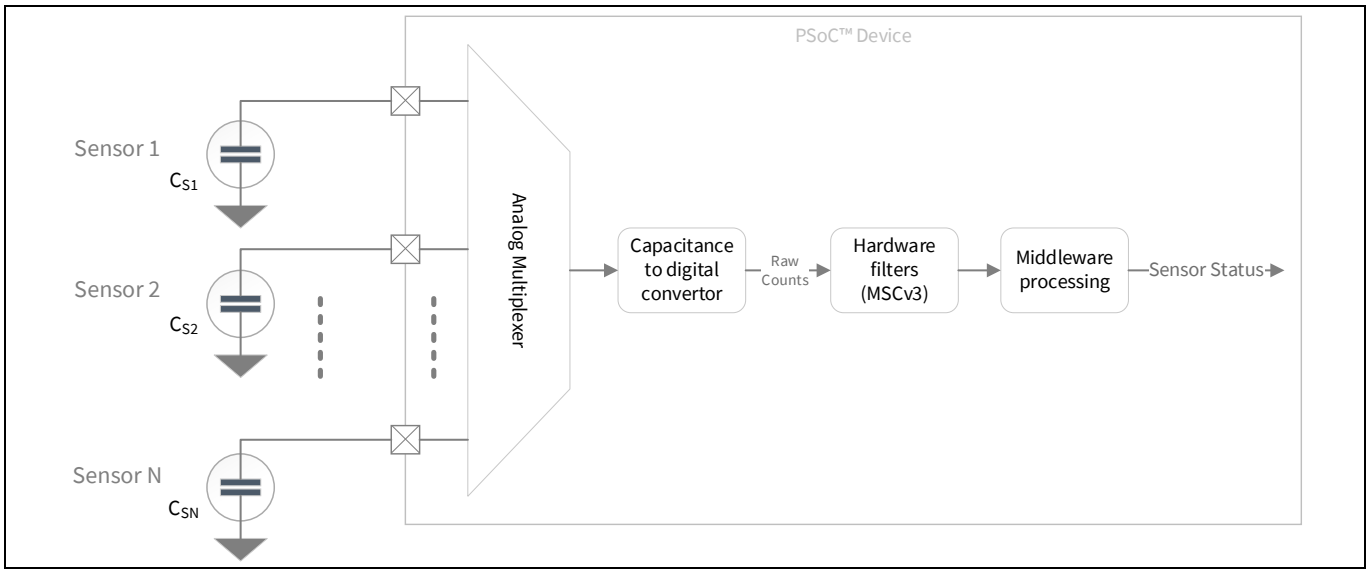


Figure 5　簡略化された CSD 変調器

CAPSENSE™における raw カウントとセンサー静電容量の関係は次のとおりです。

Raw カウント = $G_C C_S$

ここで、

G_C は静電容量からデジタルへの変換ゲインです

C_S は電極の自己容量です

Figure 6 に、時間の経過に伴う raw カウントのプロットを示します。指がセンサーに近づくと、センサー容量 (C_S) が C_P から $C_P + C_F$ に増加し、raw カウントが増加します。raw カウントの変化を所定のしきい値と比較することにより、ファームウェアロジックはセンサーがアクティブである (ターゲットオブジェクトが存在する) かどうかを判断します。

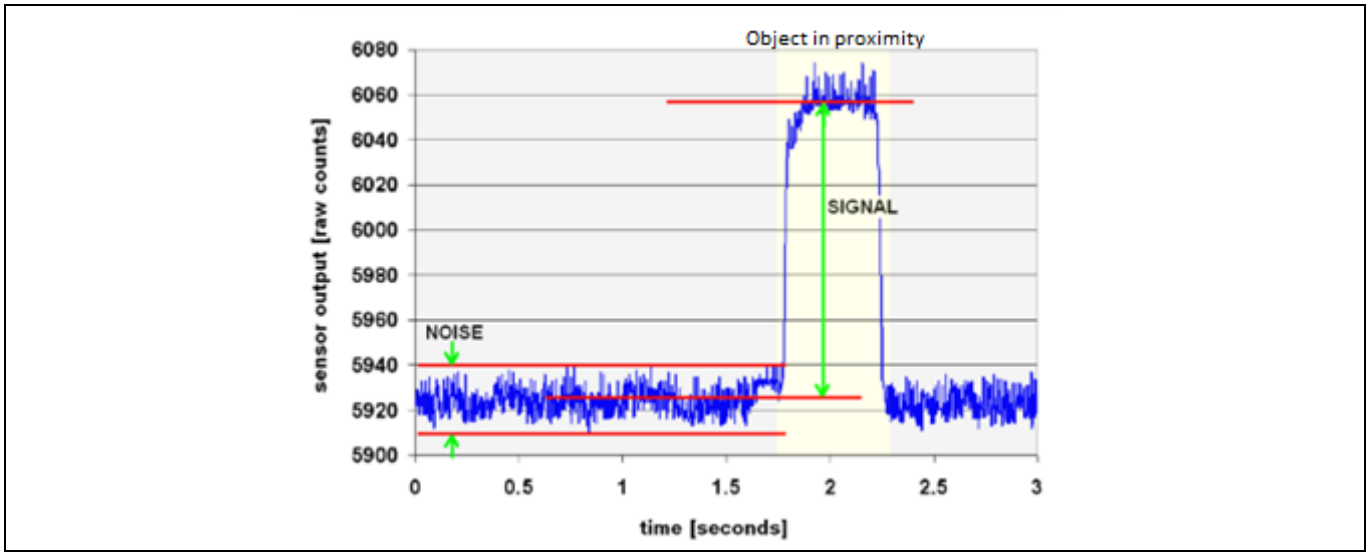


Figure 6 Raw カウントと時間の関係

1.4.3 相互容量

相互容量センシングは、2 つの近くの電極間の電気結合によって生成される静電容量を測定します。一方の電極は送信 (Tx) 電極と呼ばれ、もう一方の電極は受信 (Rx) 電極と呼ばれます。相互容量測定システムでは、デジタル電圧 (V_{DD} と GND の間で切り替わる信号) が Tx ピンに印加され、Rx 電極で受け取った電荷量が測定されます。Rx 電極で受け取られる電荷の量は、2 つの電極間の相互容量 (C_M) に正比例します。

Figure 2 に示すように、Tx 電極と Rx 電極の間に指を置くと、指による電界の妨害と指の接地効果により相互静電容量 (CM) が減少します。CAPSENSE™システムはその変化を測定します。タッチ/非タッチ状態を検出するために、Rx 電極で受信された電荷を受け取ります。

このアプリケーション ノートは、自己容量技術を使用した近接検知ソリューションの設計を説明することを目的としているため、ここで提供される相互容量の詳細は限定されています。相互容量ベースのタッチ センシング ソリューションの詳細については、AN85951 – PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™ デザインガイドを参照してください。

2 CAPSENSE™ 近接センシング ソリューションの設計

他の製品と同様に、近接検知ソリューションを設計するには、特定の要件仕様を定義する必要があります。 近接センシング設計の仕様は次のとおりです。

- 必要な最大検出距離
- 液体存在下での動作
- 極端な動作環境条件での動作
- EMI ノイズの高い場所での動作

Figure 7 に、CAPSENSE™ソリューション設計の一般的なフローを示します。このフローは、CAPSENSE™ 設計に「チューニング」と呼ばれる追加のステップが含まれることを除いて、他の電子製品の設計フローと似ています。

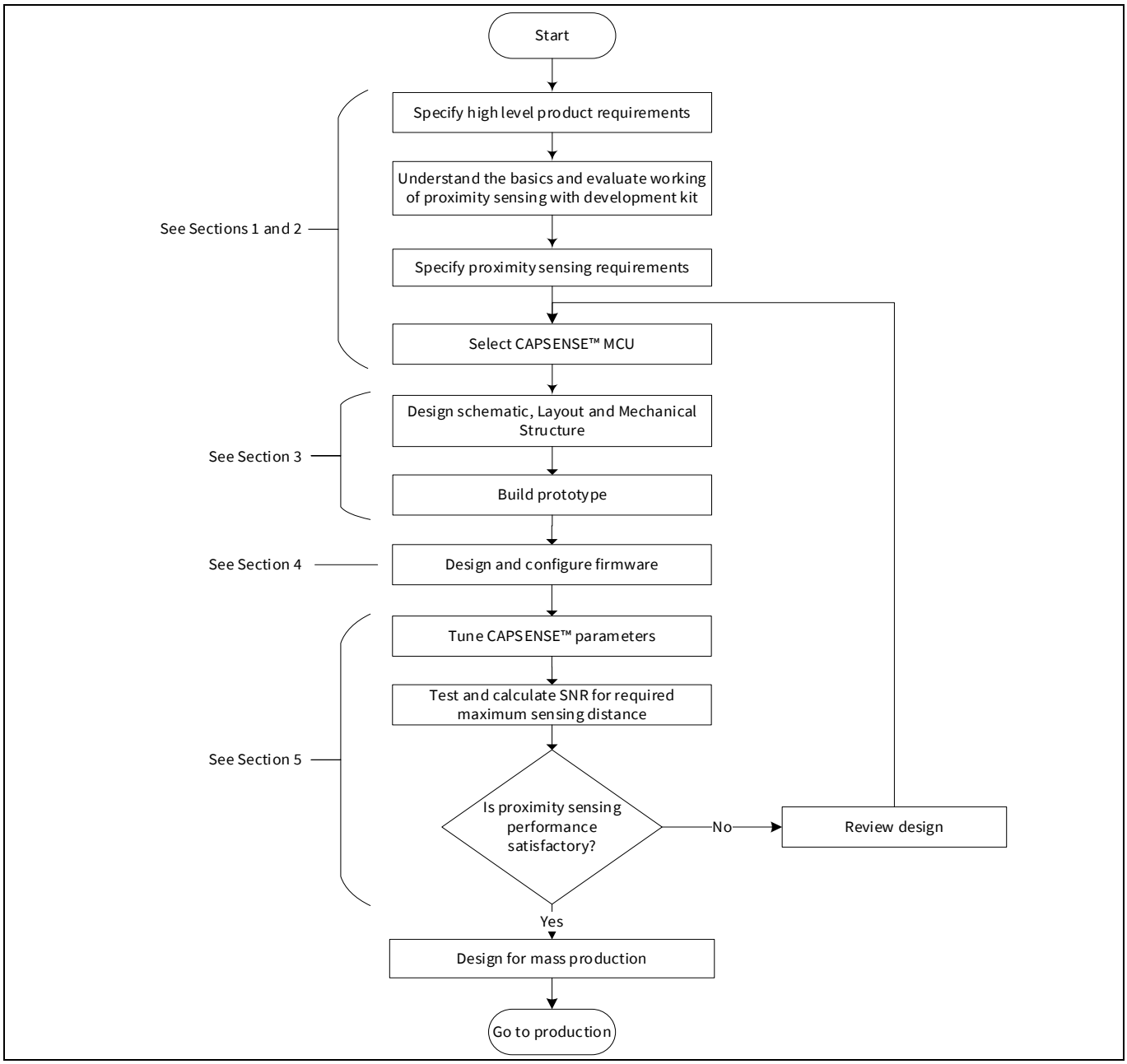


Figure 7 推奨される CAPSENSE™ 設計フロー

2.1 用語集

Table 2 に、ドキュメント全体で使用される CAPSENSE™関連用語の定義を示します。

Table 2 用語集

用語	説明
検出距離	検出距離は、追加された静電容量がいくつかのしきい値を超える距離です。検出距離はセンサーの電界伝播 (電界強度) によって異なります。伝播距離が長いほど、検出範囲も長くなります。
ターゲット オブジェクト	近接センシング設計によって存在または動きが検出されるオブジェクト (例: 人間の手/指、導電性オブジェクト)。
ウィジェット	CAPSENSE™ウィジェットは、1 つまたは複数の CAPSENSE™センサーの組み合わせであり、スライダーやトラックパッドなどの特定の種類のユーザー インターフェースを 1 つのユニットとして表します。 ウィジェットは大きく 4 つのカテゴリに分類されます。 a. ボタン (ゼロ次元) b. スライダー (1 次元) c. タッチパッド/トラックパッド(2 次元) d. 近接センサー
チャネル	PSoC™ MCU デバイス内の CAPSENSE™ペリフェラルの各インスタンスは 1 つのチャネルとみなされ、複数のインスタンスは複数のチャネルを意味します。
Raw カウント	前述したように、センサーの静電容量は CAPSENSE™ハードウェアによってカウント値に変換されます。変換されたデジタルカウント値は raw カウントと呼ばれます。raw カウントを処理すると、センサーの ON/OFF 状態が生じます。
ベースライン	センサー上に導電性の物体や人間の指が存在しない場合の raw カウントの傾向を推定するファームウェア アルゴリズムから得られる値。ベースラインは、raw カウントの突然の変化の影響をあまり受けず、差分カウントを計算するための参照点を提供します。
ベースライン リセット	ベースライン (raw カウント) 値が新しい値に設定されるイベント。ベースラインは、raw カウントがノイズしきい値内にある場合に、現在の raw カウントに基づいて再計算される値です。
差分カウントまたは信号	raw カウントからベースライン レベルを減算すると、決定プロセスで使用される差分カウントが生成されます。しきい値は、ベースラインレベルから一定量だけオフセットされます。
LP-AoS モード	第 5 世代 CAPSENSE™搭載のデバイスの低電力常時オン センシングモード。LP-AoS モードのデバイスは、ディープスリープモード中にウィジェットをスキャンして処理できます。この機能は、タッチ検出またはタイムアウト時にデバイスをウェイクアップします。
寄生容量, C_P	寄生容量、C_P 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, 回路基板を通るビア, および CAPSENSE™コントローラーのピン容量。
スキャン スロット	スキャンスロットは、一緒にスキャンされるセンサーのグループを表します。シングルチャネル モードでは、スキャンスロットごとに 1 つのセンサーがスキャンされます。

用語	説明
信号対ノイズ比 (SNR)	名前が示すように、信号内の情報の品質と信号内に存在するノイズの尺度です。容量性近接検知システムでは、SNR は信頼性が高く正確な検出を意味します。
シールド (シールド電極)	センサー周囲のハッチフィル (Figure 14 を参照) など、PCB 上の銅領域のいずれかが駆動されるシールド信号に接続されている場合、それはシールド電極と呼ばれます。

2.2 課題

静電容量型近接センシングの堅牢で信頼性の高い現実世界のアプリケーションを設計する際には、いくつかの課題を考慮する必要があります。このアプリケーション ノートでは、次のように重要な課題のほとんどに対処しようとしています。

2.2.1 低電力

低電力組込み設計は、消費電力を最小限に抑え、頻繁なバッテリ充電を必要とせずに、バッテリ駆動のアプリケーションを可能な限り長時間実行する必要性を動機としています。さらに、電力が低いということは、運用コストが低くなり、バッテリのサイズが小さくなることを意味します。

ここでは、CAPSENSE™近接センシング設計で消費電力を削減するために使用できる推奨テクニックのいくつかについて説明します。

低電力 MCU の使用

消費電力を削減する最も簡単な方法は、動作中の消費電力が非常に低いデバイスを使用することです。可能な限り最小の動作電流で設計要件を満たすデバイスを選択するには、インフィニオン Web サイトの CAPSENSE™コントローラーのページを参照してください。

第 5 世代 CAPSENSE™ MSCLP を搭載したインフィニオンの PSoC™ 4000T は、ディープスリープモードでの常時オンの容量性センシングを特長とする最も消費電力の低い MCU です。低電力モードで消費電力を最大 10 倍削減し、アクティブモードで消費電力を最大 5 倍削減できる PSoC™ 4000T は、低電力設計に推奨されます。

ウェイクオンアプローチによる低電力モードの使用

消費電力を削減する方法の 1 つは、MCU に低電力モードを実装し、ウェイク オン アプローチなど、必要な場合にのみ通常/アクティブ モードで動作することです この手法は、第 5 世代 CAPSENSE™ MSCLP デバイスが提供する低電力ウィジェットを使用して実装できます。詳細については、ウェイクオンアプローチセクションを参照してください。

MCU がアクティブなときにスキャン時間を短縮すると、デバイスが CPU をアクティブに保つのに費やす合計時間が短縮され、消費電力が削減されます。スキャン時間は、次の手法を使用して短縮できます。

- (CAPSENSE™) 変調器クロック周波数の増加
- スキャン解像度を下げる

消費電力を削減するためにこれらのパラメーターを調整する方法の詳細については、近接センシング設計の調整セクションを参照してください。

電力を節約するもう 1 つの手法は、センサー ギャングを使用したウェイク オン アプローチの実装です。これは、すべての静電容量センサーを 1 つの近接センサーとしてスキャンすることを指します。詳細については、センサーの連動セクションを参照してください。

回路の一部をオフにする

製品の消費電力を削減する手法の 1 つは、必要のないときに回路の一部をオフにすることです。このような技術では、回路の一部をファームウェアによって無効にできるように回路を設計する必要があることに注意してください。

関連リソース

CAPSENSE™ベースの設計ソリューションにおける消費電力の削減の詳細については、次のリソースを参照してください。

アプリケーションノート:

- AN234231 - Achieving lowest-power capacitive sensing with PSoC™ 4000T
- AN85951 - PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™デザインガイド

サンプルコード:

- PSoC™ 4: MSCLP CAPSENSE™ low-power proximity tuning
- PSoC™ 4: MSCLP CAPSENSE™ low power

2.2.2 長い近接センシング距離

近接センシング距離は複数の相互依存要因に依存するため、エンドシステムで大きな近接センシング距離を達成することは困難です。

Table 3 に、近接センシング距離に影響を与えるハードウェア, ソフトウェア, およびシステムパラメーターを示します。

Table 3 近接センシング距離に影響を与える要因

パラメーター	影響	改善方法/ベストプラクティス
センサーの種類とサイズ	ただし、センサーの表面積が増加すると C_P も増加し、センシング距離に悪影響を及ぼします。	塗りつぶしパターンの代わりに、可能な最大サイズのループ (またはリング) センサーを使用してください。詳細については、レイアウト ガイドラインを参照してください。
センサーの寄生容量 (C_P)	C_F/C_P 比が増加すると、近接センシング距離が長くなります。	広い検出面積を維持しながら C_P を最小限に抑えることができるようにセンサーを設計してください。詳細については、レイアウト ガイドラインを参照してください。
近くにある浮遊または接地された導電性物体	導電性の物体は電界の一部を吸収し、伝播距離が減少し、その結果、検出範囲が減少する可能性があります。	a. 導電性物体をセンサーからできるだけ遠ざけてください。 b. 物体とセンサーの間にシールドを使用するか、 c. 可能であればオブジェクトを盾として使用してください。
センサーの ADC 分解能	ADC の分解能が高いほど感度が高くなり、検出距離も長くなります。	ADC 分解能を利用可能な最大値に設定してください。詳細については、チューニングガイドラインを参照してください。

パラメーター	影響	改善方法/ベストプラクティス
近くの電気ノイズ源	電気ノイズが近接センサーの電界に干渉すると、SNR が低下し、その結果、近接センシング距離が短くなります。	• ノイズ源がセンサーからできるだけ遠ざかるように製品を設計してください。 • センサーを接地配線で囲んでください。 • フィルターは、センサーの raw カウントのノイズを軽減し、SNR を向上させるのに役立ちます。詳細については、フィルターセクションを参照してください。

長い近接センシング距離を必要とする新規の設計には、CAPSENSE™ MSCLP を備えたデバイスが推奨されます。まったく新しいレシオメトリックアナログアーキテクチャと高度なハードウェアフィルタリングに基づいて感度が向上しました。

2.2.3 耐液性

近接センサーは、蛇口やソープディスペンサーなどの用途に使用されます。これらのアプリケーションでは、誤トリガーを引き起こす可能性のある水滴やその他の液体が存在する場合でも、堅牢な動作が必要です。

設計を耐液性にするには、設計のあらゆる段階 (レイアウト, ファームウェア, チューニング) でさまざまな考慮事項を行う必要があります。CAPSENSE™ 設計を耐液性にするための推奨戦略は次のとおりです。

- シールド電極の有効活用
- ガードセンサーの使用
- 耐液性の調整

詳細については、耐液性に関するレイアウトのガイドライン、耐液性に関するファームウェアのガイドライン、および耐液性のチューニングを参照してください。

関連リソース

次のサンプルコードは、CAPSENSE™ ベースの設計で耐液性を実現する方法を示しています。

- PSoC™ 4: CAPSENSE™ MSCLP Liquid tolerant proximity sensing
- PSoC™ 4: MSCLP robust low-power liquid-tolerant CAPSENSE™

2.2.4 高いセンシング信頼性/耐ノイズ性

近接センサーはセンサー面積が大きく、感度設定が高いため、ノイズの影響を受けやすくなります。ノイズが高いと、信頼性の高い近接センシングに必要な良好な信号対ノイズ比 (SNR) (通常は 5:1 以上) を達成することが困難になります。

Table 4 に、近接センシングにおけるノイズの原因となる一般的な原因と、それぞれの推奨される緩和手法を示します。

Table 4 ノイズの発生源

ノイズの発生源	例	推奨されるノイズ軽減手法
PWM 駆動デバイス	LED、モーター	これらの発生源をセンサーからできるだけ遠ざけるように設計してください。これらのデバイスとセンサーの間に地上ハッチを追加します。
スイッチングパワーコンバーター	AC-DC、DC-DC、DC-AC	

ノイズの発生源	例	推奨されるノイズ軽減手法
高速通信インターフェース	USB、Ethernet	シールド付きツイストペア線を使用し、これらのソースをセンサーからできるだけ遠ざけるように設計してください。これらのデバイスとセンサーの間に地上ハッチを追加します。
AC 電源ライン	リレーおよびスイッチ	電力線がセンサーの近くを通過する場合は、シールド付きツイストペア線を使用してください。

フィルターの使用

CAPSENSE™ベースの設計でノイズを低減する最も効果的な方法は、フィルターを使用することです。フィルターは、raw カウントのノイズを大幅に削減し、SNR とセンシング精度を向上させるのに役立ちます。CAPSENSE™エコシステムによって提供されるさまざまなフィルターについては、フィルターセクションで説明します。

第 5 世代 CAPSENSE™を搭載したインフィニオンの PSoC™ 4000T デバイスは、まったく新しいレシオメトリックアナログアーキテクチャと高度なハードウェアフィルタリングに基づいて SNR を向上させ、信頼性が高く堅牢な容量性近接センシングを可能にします。

2.2.5 方向性

近接センサーは、一定の範囲内であらゆる方向から物体を検出できます。エンド システムに方向性近接センシングを実装することは、筐体全体の設計、ハードウェア コンポーネント、PCB レイアウトなどの複数の要因に依存するため、大きな課題となります。

近接センサーで方向性感度を実現するには、Figure 8 に示すように、センサーの下部にグランド プレーンを配置します。グランド プレーンはセンサーの感度を低下させる可能性があるため、シールドからある程度離して配置する必要があります。最適な距離はさまざまなシステム要因によって異なり、最適な距離を決定するには実際のシステムでテストする必要があります。

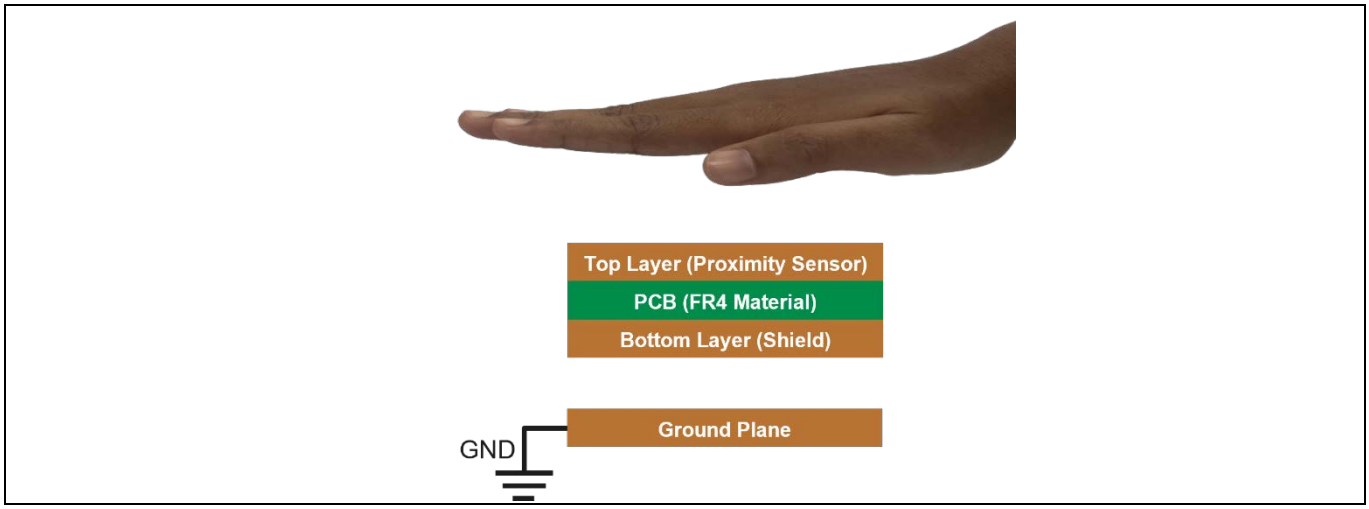


Figure 8 近接指向性のためのグランドプレーン配置

2.3 CAPSENSE™ 近接センシングを開始するためのリソース

ここでは、CAPSENSE™近接センシングソリューションの設計を開始するためにインフィニオンが提供するリソースについて説明します。

2.3.1 オンラインリソース

- Infineon CAPSENSE™ webpage
- Infineon ModusToolbox™ webpage
- ModusToolbox™ software help on GitHub

2.3.2 入門キット

1. CY8CKIT-040T PSoC™ 4000T CAPSENSE™評価キット

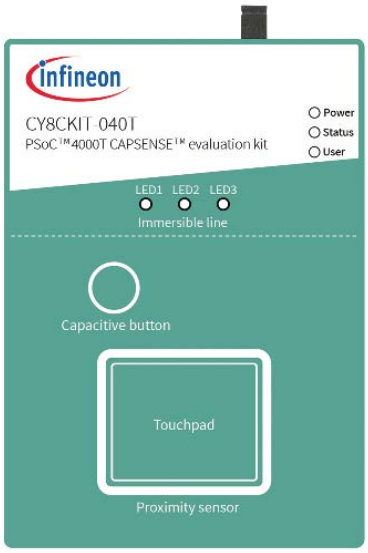


Figure 9 CY8CKIT-040T PSoC™ 4000T CAPSENSE™ 評価キット

CY8CKIT-040T PSoC™ 4000T CAPSENSE™ Evaluation Kit は、PSoC™ 4000T シリーズ MCU out-of-the-box (OOB) で利用可能な第 5 世代 CAPSENSE™テクノロジーの次の主要な機能を実証します。

- 優れたタッチおよび近接センシング性能
- 「常時オン」センシングに基づく超低電力機能
- 優れた耐液性

2. CAPSENSE™ 近接シールド

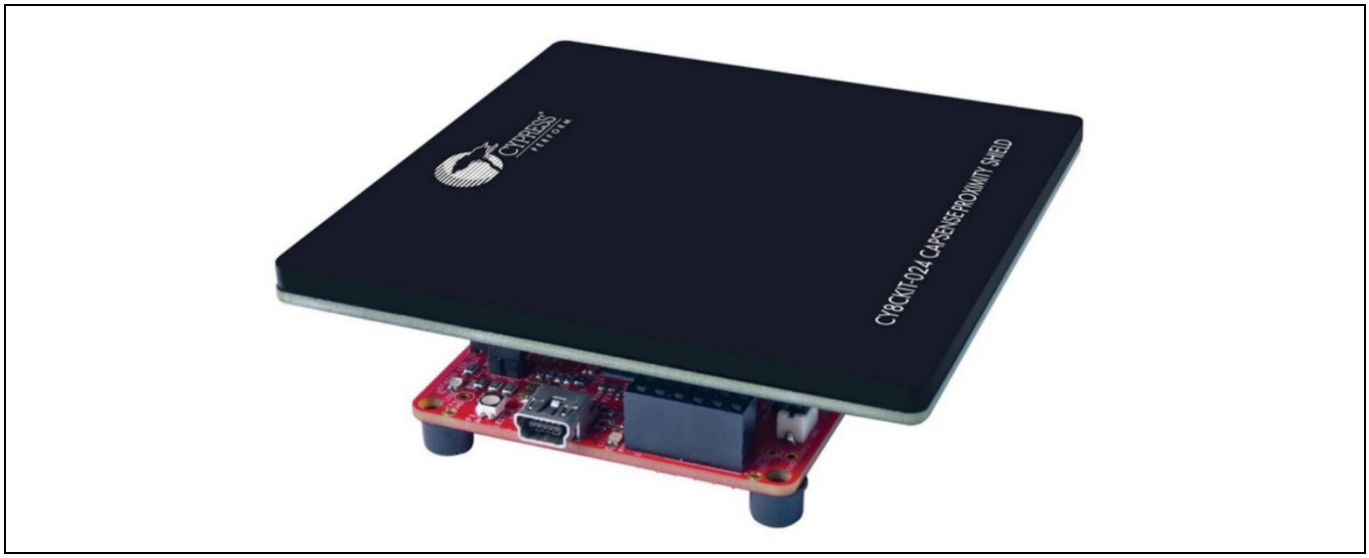


Figure 10　CAPSENSE™近接シールドを備えた CY8CKIT-024

この Arduino 互換の CAPSENSE™近接シールドを備えた CY8CKIT-024 は、PSoC™製品における CAPSENSE™ テクノロジーの近接センシング機能を実証します。CY8CKIT-042 PSoC™ 4 Pioneer Kit や CY8CKIT-040 PSoC™ 4000 Pioneer Kit などの PSoC™パイオニアキットと一緒に使用できます。

2.3.3　サンプルコード

- PSoC™ 4: MSCLP CAPSENSE™ low-power proximity tuning

このサンプルコードは、CY8CKIT-040T PSoC™ 4000T CAPSENSE™評価キットを使用して、遠距離にあるターゲットオブジェクト (手) を検出する低電力近接センシングアプリケーションの実装を示します。

これには、推奨される電源状態, 遷移, チューニングパラメーターの調整, およびチューニング方法が含まれます。この例では、MSCLP 低電力ウィジェットを使用して、低電力設計を実装するためのさまざまな考慮事項を示します。

このサンプルコードでは、CSD-RM センシング技術と CAPSENSE™ Tuner を使用して、消費電力や応答時間などのパラメーターに関して最適なパフォーマンスと最大距離を実現するために低電力ウィジェットを手動で調整する方法も説明します。

- PSoC™ 4: CAPSENSE™ MSCLP liquid tolerant proximity sensing

このサンプルコードは、CY8CKIT-040T PSoC™ 4000T CAPSENSE™評価キットを使用した、耐液性を備えた低電力近接センシングの実装を示します。

3 レイアウトガイドライン

センサーのレイアウトは、必要な近接検知距離と検出の信頼性を達成する上で重要な役割を果たします。静電容量型近接センサーは、アプリケーションの要件により異なった材料を使用して、さまざまな形状、サイズで構築されます。センサーレイアウト設計時の基本的な目的は次のとおりです。

- 高感度を実現
- 信頼性の高い検出
- 最大近接センシング距離
- 最小 C_P

検出の基本原理は同じですが、(近接) ジェスチャ検出の動作は近接存在検出とは異なります。上下のジェスチャ (センサーに手を近づけたり遠ざけたりするなど) を除いて、ジェスチャ検出には複数のセンサーが必要です。したがって、近接存在検出とジェスチャ検出ではセンサーのレイアウト設計が異なりますので、ここで説明します。

3.1 存在検出

前述したように、近接存在センシングは、物理的接触を行わずに近くのオブジェクトの存在を検出するプロセスです。Table 5 は、静電容量型近接センサー設計の考えられるタイプとそれぞれの使用例を示しています。

Table 5 静電容量型近接センサーのタイプ

センサーのタイプ	推奨される使用例	センサーC_P	推定検知距離
ボタンセンサー	近接センシングが必要な場合、センサーが利用できる距離または領域は非常に狭い。	高	~センサーの直径に等しい*
連動センサー	PCB 上に近接センサーの実装に使用できる専用のセンサー ピンまたは領域がないが、静電容量式タッチ センサーは使用できる場合。	最高	感知範囲は、連動するセンサーの種類と数によって異なる
PCB トレース	必要な近接センシング距離が非常に長い場合。	オプション	~センサーの直径/対角に等しい*
ワイヤーループ	必要な近接センシング距離が非常に長く、PCB 上に十分な領域が利用できない場合、または PCB トレースタイプセンサーを妨げるその他の設計上の制約がある場合。	オプション	~センサーの直径/対角に等しい*

* 実際のセンシング距離は、長い近接センシング距離で説明した複数の要因に影響されます。ここで記載されている検知距離は推定に基づいています。

3.1.1 ボタンセンサー

設計の唯一の目的が近接センシングである場合、ボタンセンサーは推奨されません。一般に、ボタンセンサーを使用する理由は、既存のタッチセンサーが近接センシングに再利用されるという設計上の制約です。

そのため、ボタンセンサーは (固体の) 表面積で構成されており、高い C_P を示し、その結果、検出解像度と有効検出範囲が低くなります。また、ボタンセンサーの直径は通常 5 mm ～ 15 mm であるため、ボタンセンサーで達成できる近接センシング距離は、他のセンサー実装方法と比較すると制限されます。

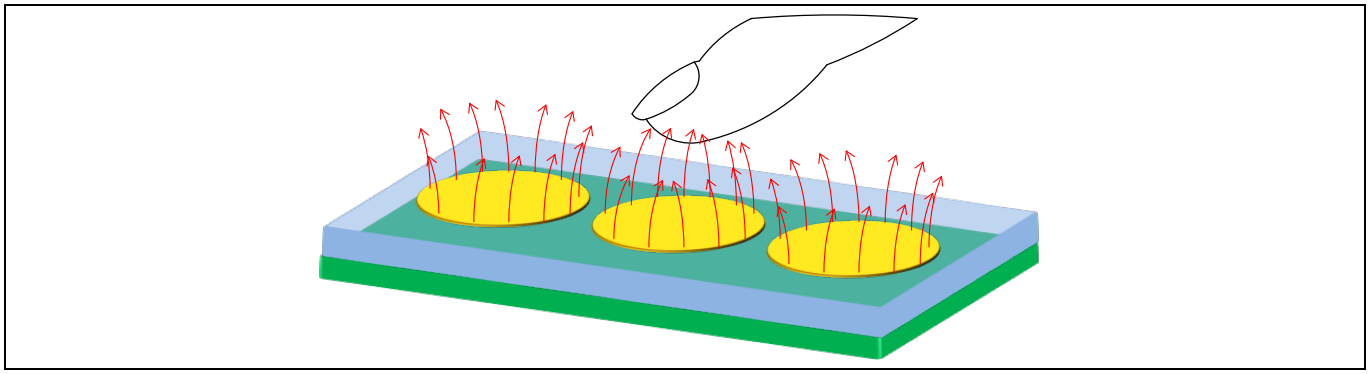


Figure 11　ボタンセンサーによる近接センシング

単純なボタンセンサーで実現できる距離よりも大きな距離の近接センシングが必要なアプリケーションでは、ボタンの周囲にリングを付けて別個の近接センサーとして機能させることを推奨します。センサー電極の底部にシールド電極を配置するなど、他の距離改善方法と併用すると、検出距離を改善できます。

3.1.2　連動センサー

センサーの連動とは、複数のセンサー (ボタン, 近接トレース, 近接ループなど) を CAPSENSE™回路に接続し、単一のセンサーとしてスキャンすることを指し、通常はウェイクオンアプローチ手法に使用されます。 詳細については、センサーの連動セクションで説明します。

3.1.3　PCB トレースセンサー

FR4 またはフレキシブル プリント回路 (FPC) 基板上の長い PCB 配線は、近接センサーを形成する可能性があります。トレースは直線 (Figure 12 (a)) にすることも、Figure 12 (b) に示すようにシステムのユーザー インターフェースの周囲を囲むこともできます。PCB トレースを使用した近接センサーの実装には、他のセンサー実装方法と比較した場合、次の利点があります。

- 低い C_P
- ループセンサーの場合、より多くの電界線が手/導電性の物体に結合するため、近接センシング距離が長い
- 量産に適する

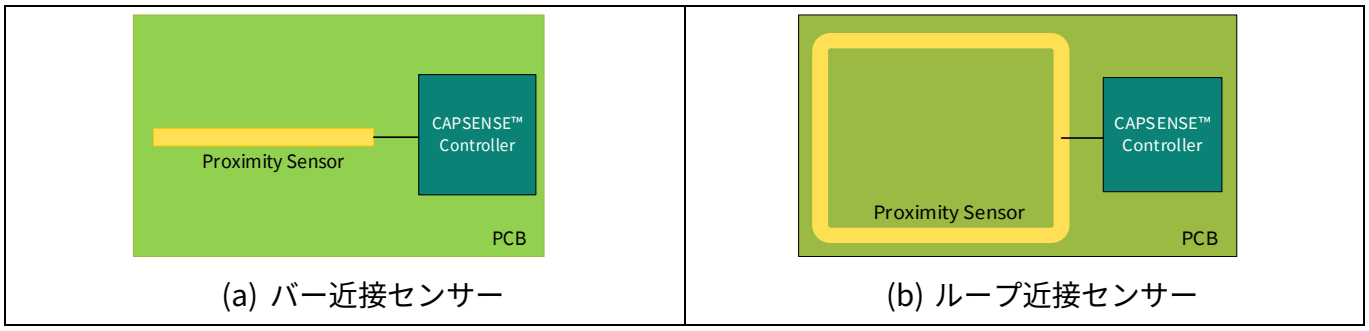


Figure 12　PCB トレースを使用した CAPSENSE™ ベースの近接センシング

3.1.4 ワイヤーセンサー

Figure 13 に示すように、単一の長さのワイヤーをループバックすると、近接センサーとして機能します。ワイヤーループを使用した近接センサーの実装には、他のセンサー実装方法と比較して次の利点があります。

- 製品内のセンサーの設置場所の柔軟性が向上します。例えば、ワイヤーループを製品のケーシング/筐体の壁の下に取り付けることができます。
- 実装サイズに柔軟性があるため、より大きな近接距離を実現できます。

ただし、ワイヤーセンサーの使用は、製造コストと複雑さのため、大量生産には最適なソリューションではありません。したがって、推奨できません。

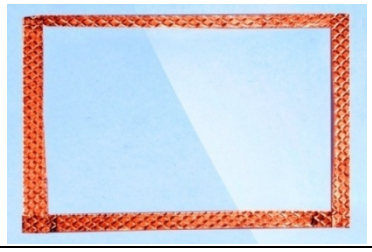


Figure 13 ワイヤーループとして銅テープを使用した近接センサーのプロトタイプ

3.2 一般的なレイアウトのガイドライン

ここでは、あらゆるタイプの静電容量型近接センサーの構築に適用されるガイドラインについて説明します。これらのガイドラインは、静電容量型近接センシング設計のパフォーマンスを向上させるために推奨されます。

3.2.1 センサーのサイズと近接センシング距離

センサー面積が大きくなると、対象物体と結合する電力線が増加し、センサー信号が増加します。したがって、必要なセンサーのサイズは、設計に必要な近接センシング距離に応じて増加します。ただし、センサー面積が大きいと、センサーの C_P が高くなり、ノイズが高くなる可能性があります。これは、近接センシング距離に悪影響を及ぼします。固体充填センサーの代わりに等価直径/対角のループセンサー (Figure 12 (b)) を使用すると、センサーC_P が低くなり、ノイズが減り、近接検知距離が向上します。

近接センサーは本質的に高感度であるため、近くのノイズ源や浮遊または接地された導電性物体により SNR が低下し、その結果、近接センシング距離が低下します。

経験則として、必要な近接センシング距離に等しい最小ループ直径 (円形ループの場合) または対角線 (正方形ループの場合) から始めることを推奨します。必要な近接センシング距離と等しいループ直径または対角線で必要な近接センシング距離を達成できない場合は、必要な近接センシング距離が達成されるまでセンサーサイズを大きくしてください。

3.2.2 センサーの寄生容量

C_F はターゲットオブジェクトの存在下での静電容量の変化を示すため、近接センシング距離は測定分解能に影響するため、C_F と C_P の比率に依存します。近接センシング距離は、C_F/C_P 比が増加すると増加します。特定のセンサーサイズの場合、C_F の値はセンサーとターゲットオブジェクト間の距離とターゲットオブジェクトのサイズによって異なります。C_F は設計によって完全に制御できるわけではありませんが、C_F/C_P 比を最大化するには、カバーされる検出領域を同等に保ちながら C_P を下げることが最も簡単なアプローチです (ループ センサーなど)。

センサーの C_P は次のように最小化できます。

- 最適なセンサータイプの選択
- センサー配線長の短縮
- センサーの電力線と近くの地面または金属物体との結合を最小限に抑える

センサーの配線長を最小限に抑え、それによってセンサーC_Pを最小限にするには、CAPSENSE™ MCU デバイスをできるだけセンサーの近くに配置します。

センサーの電界線のグランドへの結合を軽減するには、次のセクションで説明するように、PCB の上層と下層でセンサーを囲むハッチフィルを実装します。

3.2.3 シールド電極

Figure 14 に示すようなハッチフィルがセンサーの周囲に実装され、信号 (センサーのものと同じ) で駆動される場合、これはシールド電極と呼ばれます。センサーを囲む上層と下層に、0.17 mm (7 mil) トレースと 1.143 mm (45 mil) グリッドのハッチフィルを備えたシールド電極は、次の目的に使用できます。

- センサーの C_P を下げるため
- 耐水性の向上
- 近接センシング距離に対する浮遊/接地された導電性物体の影響を軽減するため
- 近接センシングを一方向にするため

Figure 14 に、センサーの C_P を削減するためのシールド設計の例を示します。

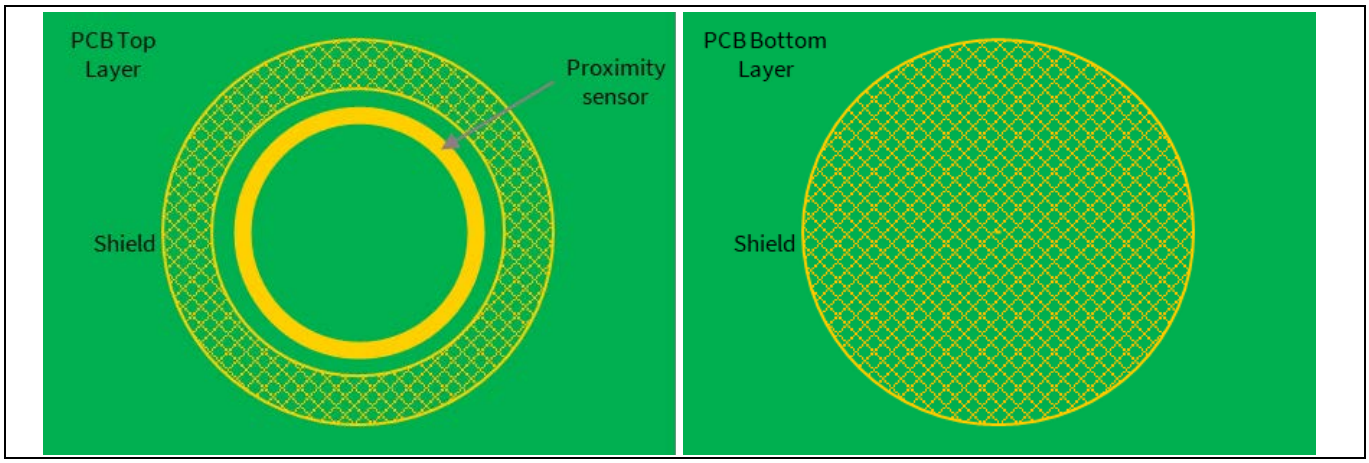


Figure 14 シールド電極のパターンと配置例

3.3 外部コンデンサ

CAPSENSE™デバイスには、自己容量センシング用の外部コンデンサが必要です。これらの外部コンデンサは、専用の GPIO ピンとグランドの間に接続されます。Table 6 に、各 CAPSENSE™世代の外部コンデンサの推奨値を示します。

Table 6　外部コンデンサの推奨値

タイプ	CAPSENSE™世代	推奨値
C_{MOD}	3rd および 4th	2.2 nF
C_{MOD1} および C_{MOD2}	5th	2.2 nF
C_{SH_TANK}	3rd および 4th	シールド電極が実装されている場合は 10 nF、 それ以外の場合は NA

詳細については、AN85951 – PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™デザインガイドの「回路図ルールチェックリスト」セクションを参照してください。

3.4　ピン配置

近接センサー トレースと通信または非センサートレース間の相互作用を減らす効果的な方法は、ポート(ピン) 割り当てによってそれぞれを分離することです。Figure 15 に、32 ピン QFN パッケージのこの絶縁の基本バージョンを示します。各機能が分離されているため、CAPSENSE™コントローラーは、通信, LED, および検出トレースが交差しないように設計されています。

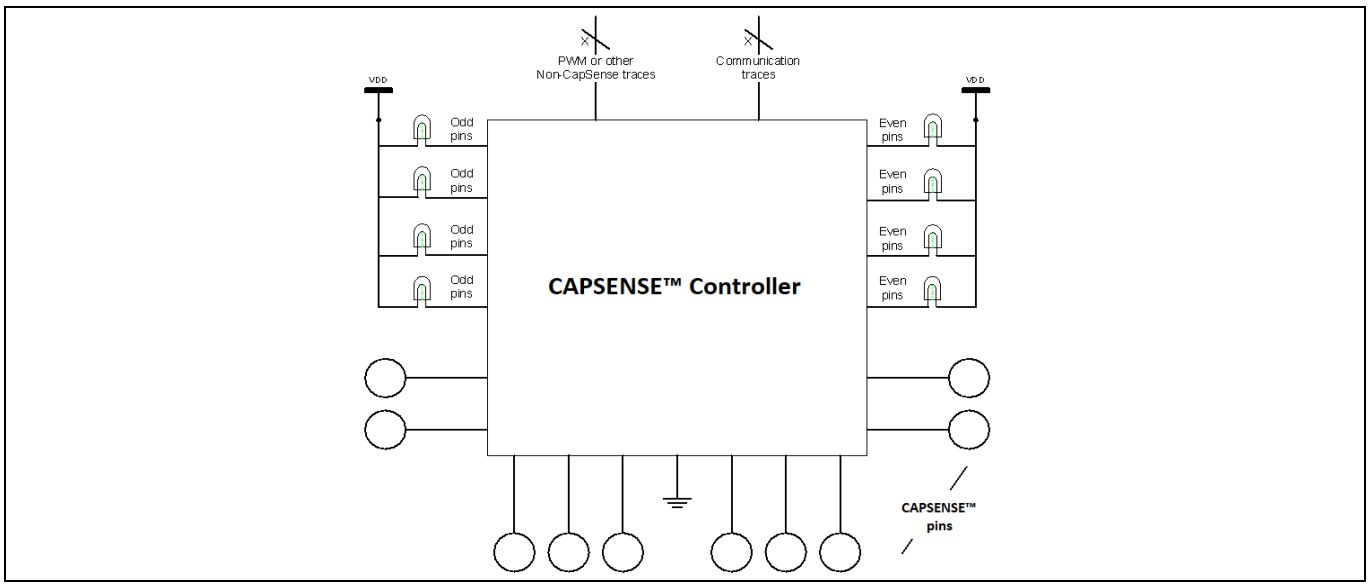


Figure 15　推奨: 通信, CAPSENSE™, および LED 用のポート分離

CAPSENSE™ コントローラーアーキテクチャは、偶数ポート ピンと奇数ポートピンの電流バジェットに制限を課します。CAPSENSE™コントローラーの場合、奇数ポートピンの電流バジェットが 100 mA の場合、すべての奇数ポート ピンが 100 mA を超えてはいけませんが、合計電流が消費されます。合計電流バジェット制限に加えて、各ポートピンの最大電流制限もあります。詳細については、アプリケーションで使用される CAPSENSE™コントローラーのデータシートを参照してください。

すべての CAPSENSE™コントローラーは、高電流シンクおよびソース対応ポートピンを提供します。ポートピンからの大電流シンクまたはソースを使用する場合は、ノイズを最小限に抑えるためにデバイスのグランドピンに最も近いポートを選択してください。

Note: CAPSENSE™がセンサーをスキャンしている間は、GPIO を流れる合計ソース/シンク電流を 40 mA に制限してください。センサーのスキャン中に 40 mA を超えるシンク/ソースを行うと、センサーの raw カウントに過剰なノイズが発生する可能性があります。

ModusToolbox™ CAPSENSE™ デバイスコンフィギュレータは、Figure 16 に示すように、CAPSENSE™ デザインのピン割り当てに使用できます。

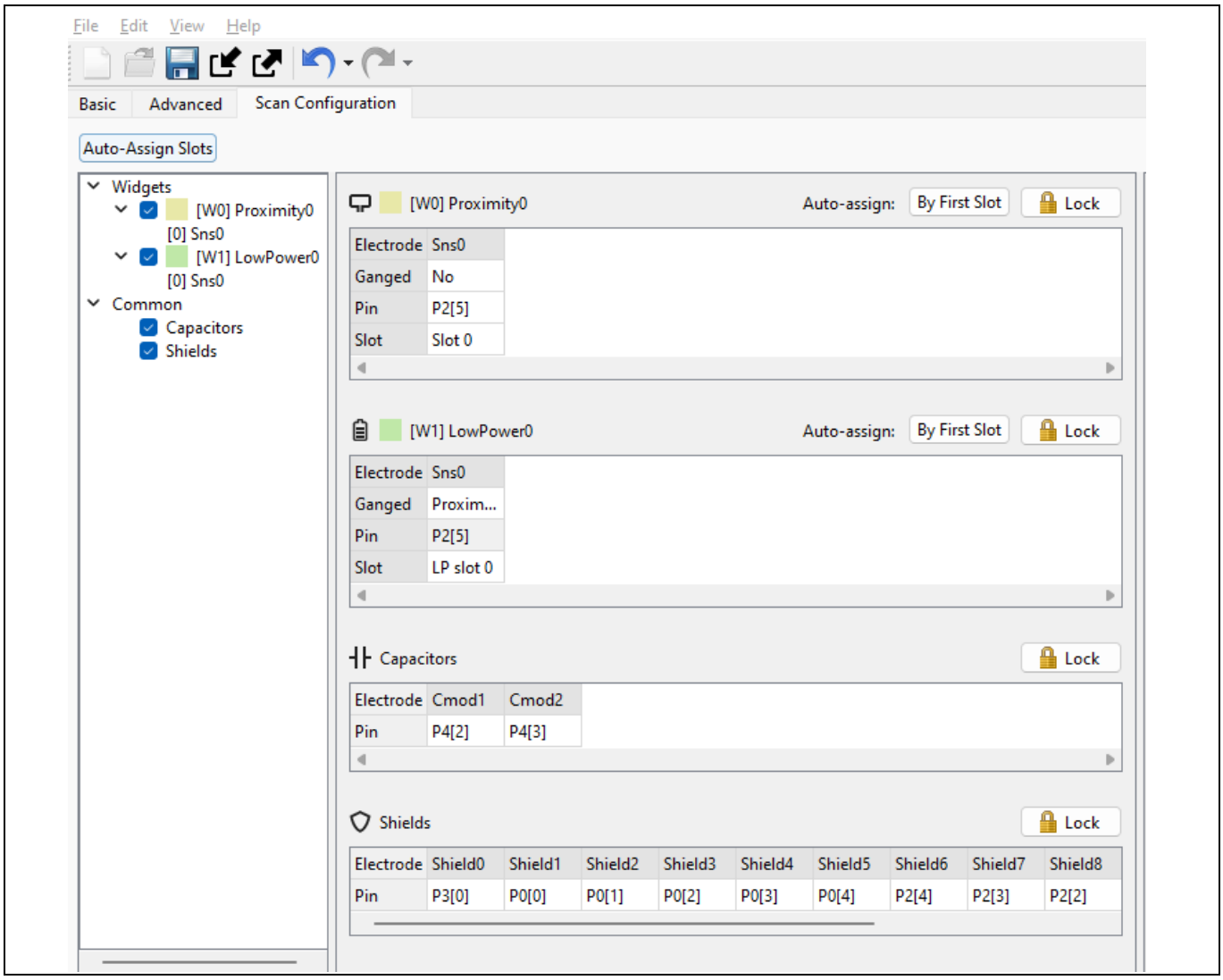


Figure 16 ModusToolbox™ CAPSENSE™コンフィギュレータを使用した PSoC™ピン割り当て

3.4.1 外部コンデンサのピン選択

Table 7 に、CAPSENSE™ベース設計の C_{MOD} および C_{SH_TANK} コンデンサの推奨ピンを示します。

Table 7 外部コンデンサの推奨ピン

デバイス	C_{MOD} (または第 5 世代 CAPSENSE™ の C_{MOD1})	C_{SH_TANK} (または第 5 世代 CAPSENSE™の C_{MOD2})
PSoC™ 4000	P0[4]	P0[2]
PSoC™ 4000T	P4[2]	P4[3]
PSoC™ 4100/PSoC™ 4200	P4[2]	P4[3]
PSoC™ 4200M/PSoC™ 4200L	CSD0: P4[2]	CSD0: P4[3]
	CSD1: P5[0]	CSD1: P5[1]
PSoC™ 4 Bluetooth® LE	P4[0]	P4[1]

デバイス	C_{MOD} (または第 5 世代 CAPSENSE™ の C_{MOD1})	C_{SH_TANK} (または第 5 世代 CAPSENSE™の C_{MOD2})
PSoC™ 6 MCU	P7[1]	P7[2]
PSoC™ 4S-series, PSoC™ 4100S Plus	P4[2]	P4[3]
PSoC™ 4100PS	P5[2]	P5[3]
PSoC™ 4100S Max	Channel0: P4[0]	Channel0: P4[1]
	Channel1: P7[0]	Channel1: P7[1]

PSoC™デバイスの外部コンデンサをサポートするピンの詳細については、それぞれのデバイスのデータシートを参照してください。

3.5 導電性物体/金属が近くにある場合の検知

浮遊または接地された導電性物体が近くにある場合、近接検知距離は大幅に減少します。導電性の物体が近接センサーの近くに配置されると、次の要因により近接センサーの距離が大幅に減少します。

- 用語集のセクションで前述したように、検出距離は電界伝播に基づいており、金属表面は電界の一部を吸収し、伝播距離が減少し、したがって検出範囲が減少する可能性があります。
- センサーの磁力線が導電性/接地物体に結合すると、センサーC_Pが増加します。センサーC_Pが大きくなると C_F/C_P 比が低下し、多くの場合センサーのスイッチング周波数を下げる必要が生じ、近接検知距離が減少します (スキャン時間を長くできない場合)。

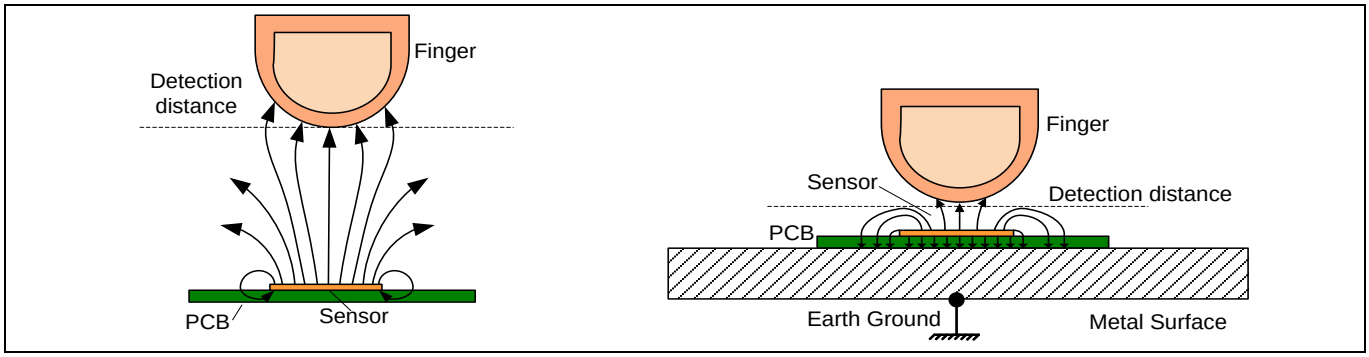


Figure 17 金属物体がある場合とない場合の電界伝播

Figure 18 に示すように、近接センサーと金属物体の間にシールド電極を配置することで、センサーに対する金属物体の影響を大幅に軽減できます。この実装には、センサー専用の別個の PCB 構造を推奨します。

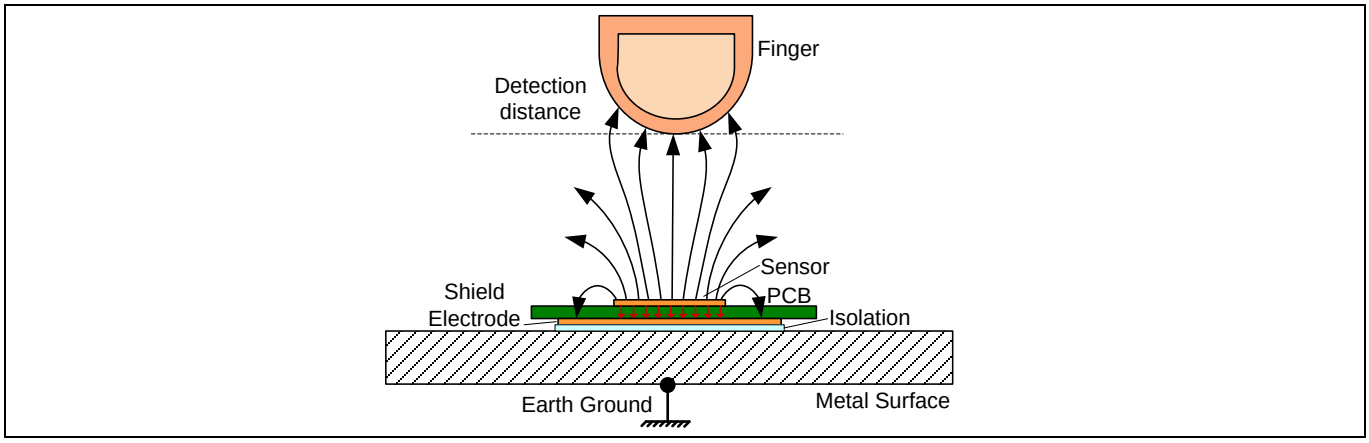


Figure 18　シールド電極を使用して金属物の影響を軽減する

設計上の制約により、PCB 上に個別の専用近接センサーを設置することができず、他のコンポーネントが取り付けられている同じ PCB にセンサーを実装する必要がある場合は、近接検知距離を改善するために次の推奨事項を実装できます。

- 検出電極を基板の周囲に配置してください。
- シールド電極は、PCB 層の最下部のセンサーの下に配置する必要があります。
- 感度低下の原因となる可能性があるので、近接センサー内の広いグランドフィル領域を使用しないでください。
- 多層 PCB を使用する場合は、最上層を 20% ～ 25%のハッチングされたシールド電極の銅の注入で満たしてください。内部層はグランドと信号のルーティングに使用できます。
- デバイスにプラスチックケースが付いている場合は、検出距離を最大化するために、内部プラスチックケース側にシールド電極を備えたワイヤーセンサーを接着してください。推奨のワイヤーの長さは 10cm～20cm です。
- シールドと金属間の距離は 10 mm ～ 20 mm としてください。

3.6　ジェスチャ検出のレイアウトガイドライン

ジェスチャを検出する場合、近接センサーの設計と配置は、検出されるジェスチャの種類によって異なります。例えば、(X 軸に沿った) 水平スワイプなどのジェスチャを検出するには、Figure 19 に示すように、水平な手の動きに対して垂直に配置された 2 つの平行な近接センサーPS1 と PS2 を使用できます。

経験則として、センサーの幅は 2 mm 未満であってはいけません。ただし、長さと幅は、センサーのサイズと近接センシング距離で説明したように、必要な検知距離によって異なります。

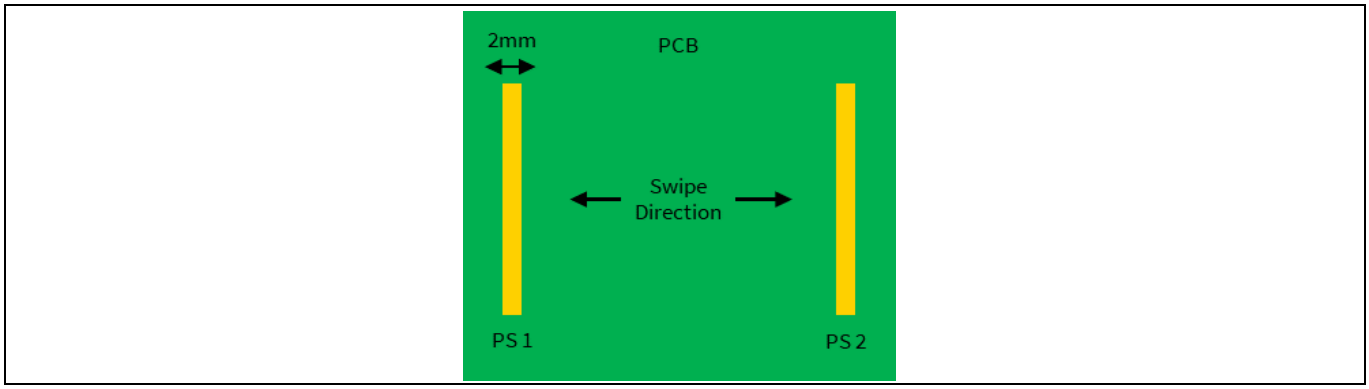


Figure 19　2 つの近接センサーを使用して単軸スワイプを検出するためのレイアウト

Figure 19 に示すセンサー配置により、Figure 20 に示す信号が生成されます。

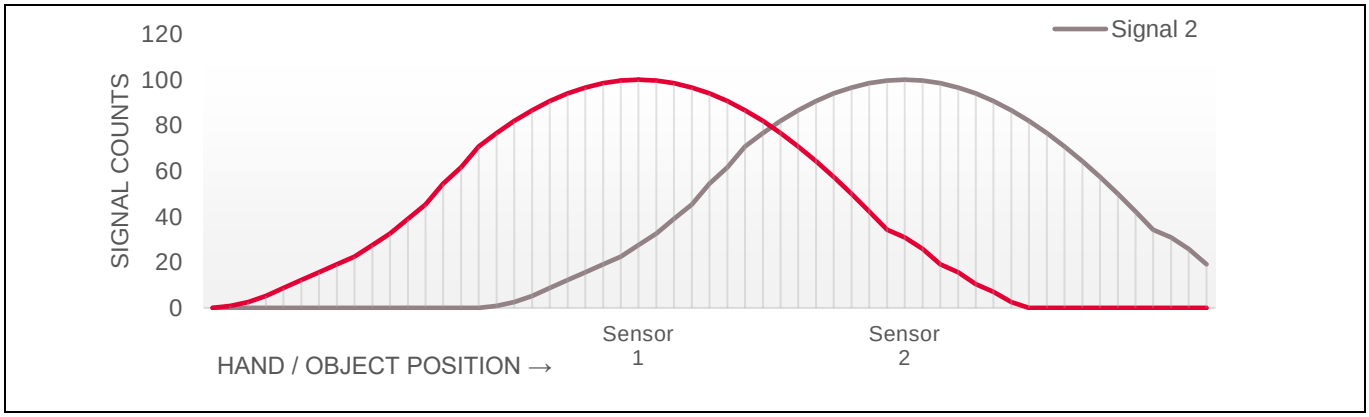


Figure 20　ジェスチャ センサー上の物体位置の信号カウント

同様に、水平スワイプと垂直スワイプ (X 軸と Y 軸の両方に沿った) を検出するには、Figure 21 に示すように 4 つのセンサーが使用されます。2 つの並列の近接センサー、PS1 と PS2 は水平スワイプ (X 軸に沿った) を検出するために使用され、PS3 と PS4 は水平スワイプ (Y 軸に沿った) を検出するために使用されます。

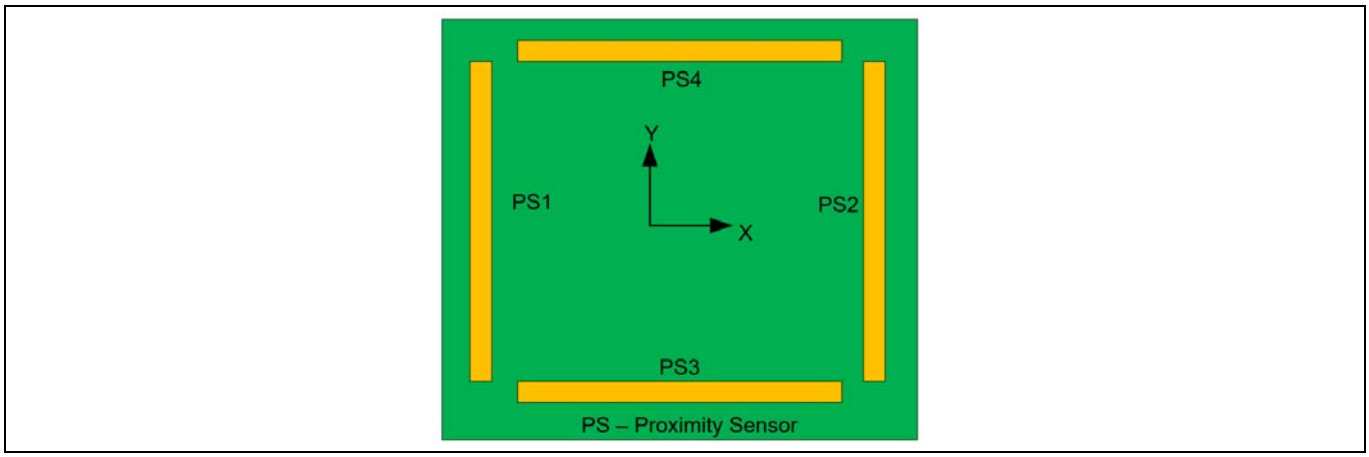


Figure 21　2D ジェスチャ検出のための近接センサーの設計と配置

複数のセンサーを使用して設計されていることを除けば、センサー設計のその他の点は単一センサー設計の場合と同じです。ジェスチャ検出センサーの設計中は、各センサーに対してすべてのレイアウトガイドラインに従うことが推奨されます。

Table 8 に、ジェスチャ検出センサーのレイアウトの設計時に考慮すべきパラメーターとそれぞれの推奨値を示します。

Table 8　　ジェスチャ検出センサーのレイアウトのパラメーターの推奨値

パラメーター	設計への影響	推奨値
トレース/センサー幅	C_p, 感度, およびセンシング距離	2 mm
トレース長さ		5 cm
トレース間の距離	検出できるジェスチャの種類	スワイプタイプのジェスチャの場合は 5 cm

3.7 耐液性に関するレイアウトのガイドライン

CAPSENSE™設計を耐液性にするには、関連するファームウェアの設定と調整とともに、レイアウト設計時にさまざまなアプローチを考慮する必要があります。耐液性 CAPSENSE™近接センシングソリューションに最も役立つ 2 つの推奨レイアウトテクニックは次のとおりです。

- シールド電極
- ガードセンサー

3.7.1 耐液近接センシング用のシールド電極の設計

慎重に設計されたシールド電極を追加すると、検出面上の液滴の干渉を大幅に軽減できます。シールド電極で説明したように、近接センサー全体内およびその周囲にシールド電極を設計します (できればハッチパターンを使用して)。センサーとシールド電極の間の最小距離を 2 mm に維持してください。

3.7.2 耐液近接センシング用のガード センサーの設計

連続的な液体の流れまたは大量の液体がセンサー表面に飛沫すると、CAPSENSE™回路に大きな静電容量が追加されます。この静電容量は C_F よりも数倍大きい可能性があります。このため、シールド電極の効果は完全にマスクされ、センサーの raw カウントが増加し、近くのターゲットオブジェクトよりも潜在的に高くなります。そんな時でもガードセンサーを使えば誤検知を防ぐことができます。

Figure 22 に示すように、ガードセンサーは、PCB 上のすべてのセンサーを囲む銅配線です。ガードセンサーはタッチセンサーに似ており、液体の流れの存在を検出するために使用されます。液体の存在を効果的に検知するには、Figure 22 に示すように複数のガードセンサーを組み合わせて使用できます。

ガードセンサーを近接センサーと組み合わせた CSX センサーとして設定します。ガードセンサーは Rx として機能し、近接センサーは Tx として機能します (近接センサーはターゲットオブジェクト検出用の CSD センサーとして、液体検出用の Tx として交互に機能します)。ガードセンサーがトリガーされると、ファームウェアはシステム内の他のすべてのセンサーのスキャンを無効にして、誤検出を防ぐことができます。この技術のファームウェアの側面については、耐液性に関するファームウェアのガイドラインセクションで詳しく説明されています。

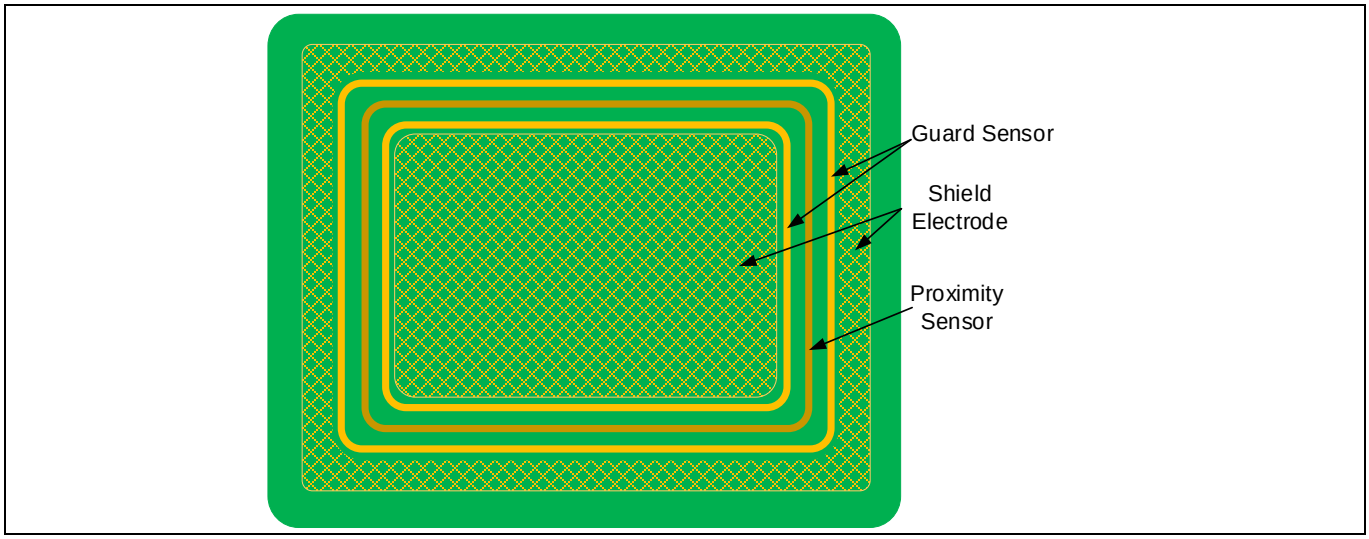


Figure 22 シールド電極とガード センサーを備えた PCB レイアウト

次のようにガードセンサーのレイアウトを設計します。

1. ガードセンサーは、エッジが湾曲した長方形の形状で、すべてのセンサーを囲んでいる必要があります。
2. ガードセンサーの推奨厚さは 2 mm です。
3. ガードセンサーとシールド電極間の推奨最小ギャップは 2 mm です。
4. センサー表面に液体がある場合、これが最初にオンになるセンサーである必要があります。これを達成するために、Figure 22 に示すように、ガードセンサーが CAPSENSE™システム内のすべてのセンサーを囲みます。
5. ターゲットオブジェクトが近くにあるときはトリガーすべきではありません (高いしきい値を使用)。そうしないと、近接スキャンが無効になり、ガードセンサーがオフになるまで CAPSENSE™システムが動作しなくなります。

PCB 上に専用のガードセンサーを実装するためのスペースがなく、設計に複数のタッチ センサーがある場合は、ガードセンサーの機能をファームウェアに実装できます。例えば、液体が存在する場合、一度に複数のセンサーがアクティブになり、さまざまなセンサーのオン/オフ状態を使用して液体の存在を検出できます。

3.8 レイアウト シミュレーション

提案された CAPSENSE™システム レイアウトのシミュレーションの詳細については、インフィニオンのテクニカルサポートにお問い合わせください。

4 ファームウェア設計ガイドライン

近接センサーのレイアウトの準備ができたら、次のステップはファームウェアの実装です。ここでは、ファームウェアと CAPSENSE™システムパラメーター設定のガイドラインについて説明します。

ModusToolbox™を使用して CAPSENSE™プロジェクトを作成する方法の詳細については、PSoC™ 4: MSCLP CAPSENSE™ 低電力近接調整サンプルコードを参照してください。

4.1 ウィジェットの種類の設定

第 5 世代 CAPSENSE™を搭載した PSoC™ デバイスは、アクティブウィジェットと低電力ウィジェットの 2 種類のウィジェット設定を提供します。

4.1.1 アクティブウィジェット

アクティブウィジェットは、CPU のアクティブモード中にスキャンされる通常のウィジェットであり、スリープモード中にはスキャンできません。第 5 世代 CAPSENSE™ デバイスによって提供される低電力ウィジェットを除き、CAPSENSE™ コンフィギュレータで使用できるすべてのウィジェットタイプは、デフォルトでアクティブなウィジェットタイプです。

4.1.2 低電力ウィジェット

PSoC™デバイスは、アクティブ, スリープ, ディープスリープなどの複数の電力モードを提供します。第 5 世代 CAPSENSE™を搭載したデバイスは、ディープスリープモードでも CAPSENSE™ペリフェラルをアクティブに保つ機能を提供します。この機能は、「低電力ウィジェット」と呼ばれるウィジェットタイプを使用して実装できます。これらのウィジェットは、ディープスリープモードでは CPU の介入なしでスキャンおよび処理されます。これらはデバイスのウェイクアップソースとして機能します。

ボタン, 連動センサー, 近接センサーなどのウィジェットを低電力ウィジェットとして設定し (Figure 23 を参照)、ウェイクオンタッチ/ウェイクオンアプローチ機能を有効にできます。

Note: ウィジェットがアクティブモードでの通常動作用の低電力ウィジェットとして設定されている場合は、さらにアクティブウィジェットとしても設定する必要があります。

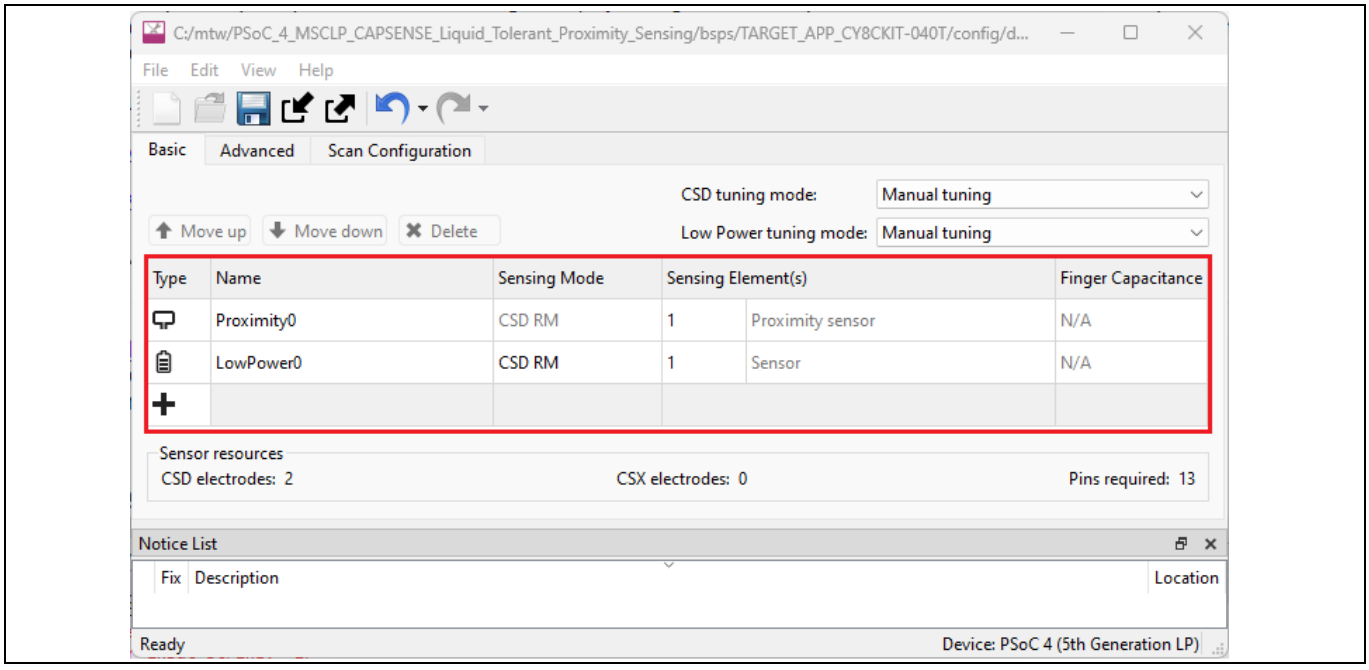


Figure 23 ModusToolbox™ の CAPSENSE™ コンフィギュレータを使用した低電力ウィジェットの追加

4.2 スキャン設定

スキャン設定では、CAPSENSE™チャネル間のセンサーの配分を定義し、連動接続, ピン割り当て, 各センサーチャネルのスキャンスロット割り当てを作成します。

チャネル

チャネルは、CAPSENSE™コンバーターのインスタンスを表します。インフィニオンは、複数のチャネルを備えた PSoC™第 5 世代 CAPSENSE™デバイスを提供しています。詳細については、Infineon Web サイトの CAPSENSE™コントローラーのページを参照してください。複数のチャネルは複数のセンサーを同時にスキャンでき、各チャネルは 1 度に 1 つのセンサーをスキャンします。ただし、ほとんどの近接検知アプリケーションでは、単一チャネルで十分です。

スロット

スロットは、(第 5 世代) CAPSENSE™の完全なスキャンサイクルのタイムスライスを表します。すべてのセンサーのスキャンがトリガーされると (第 5 世代 CAPSENSE™ の場合)、センサーは割り当てられたスロットの順序でスキャンされます。マルチチャネルモードでは、スキャンスロットは一緒にスキャンされるセンサーのグループ (各チャネルで 1 つのセンサーがスキャンされる) を表します。シングルチャネルモードでは、スキャン スロットごとに 1 つのセンサーがスキャンされます。

4.2.1 チャネルとスロットの設定

センサーのチャンネルとスロットは、CAPSENSE™コンフィギュレータの **Scan Configuration** タブを使用して割り当てることができます。詳細については、CAPSENSE™コンフィギュレータ ユーザーガイドを参照してください。

4.3 シールド設定

前述したように、センサー周囲のハッチフィルなど、PCB 上の銅領域はいずれも、シールド電極と呼ばれる信号 (センサー信号と同じ) に接続されています。

シールド電極は次の目的に使用できます。

- 耐液性の CAPSENSE™設計を実装するため
- 浮遊または接地された導電性物体が存在する場合の近接検知距離を向上させるため
- センサーの寄生容量を低減するため

第 5 世代の CAPSENSE™アーキテクチャは、次の 2 つのシールドモードに対応します。

1. アクティブシールド (ドリブンシールド)

2.パッシブシールド

シールド設定は、Figure 24 に示すように、ModusToolbox™の CAPSENSE™コンフィギュレータを使用して設定できます。

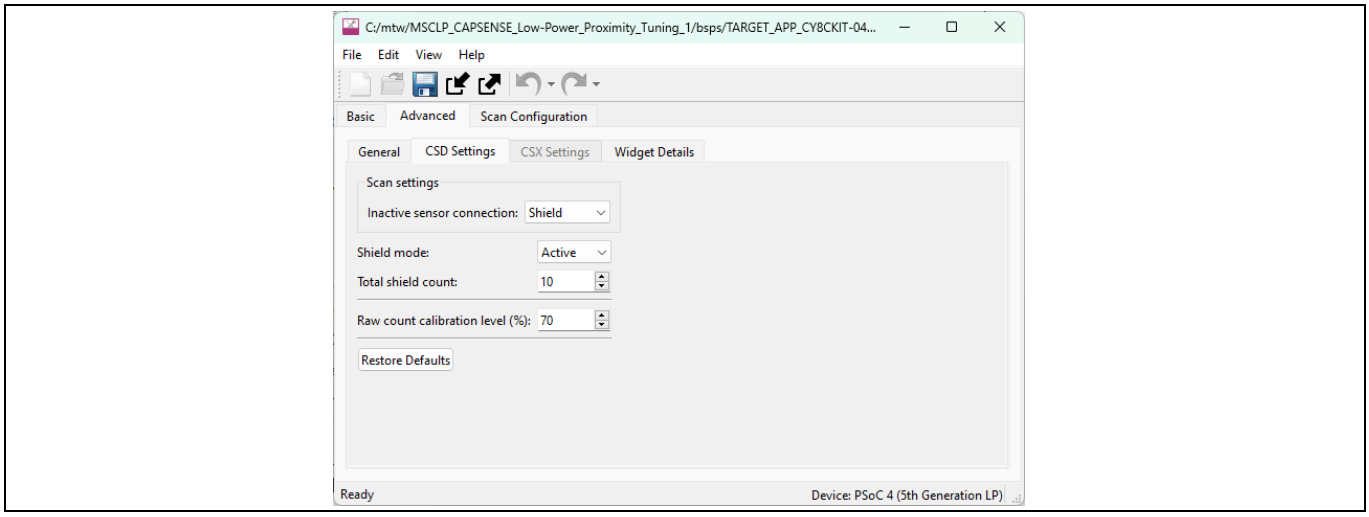


Figure 24 ModusToolbox™の CAPSENSE™コンフィギュレータでのシールド設定

4.3.1 アクティブシールド

アクティブシールドモードでは、Figure 25 に示すようにシールド回路はバッファを使用してセンサー信号のレプリカでシールド電極を駆動します。これによりセンサーとシールド電極間の電位差が無効になります。

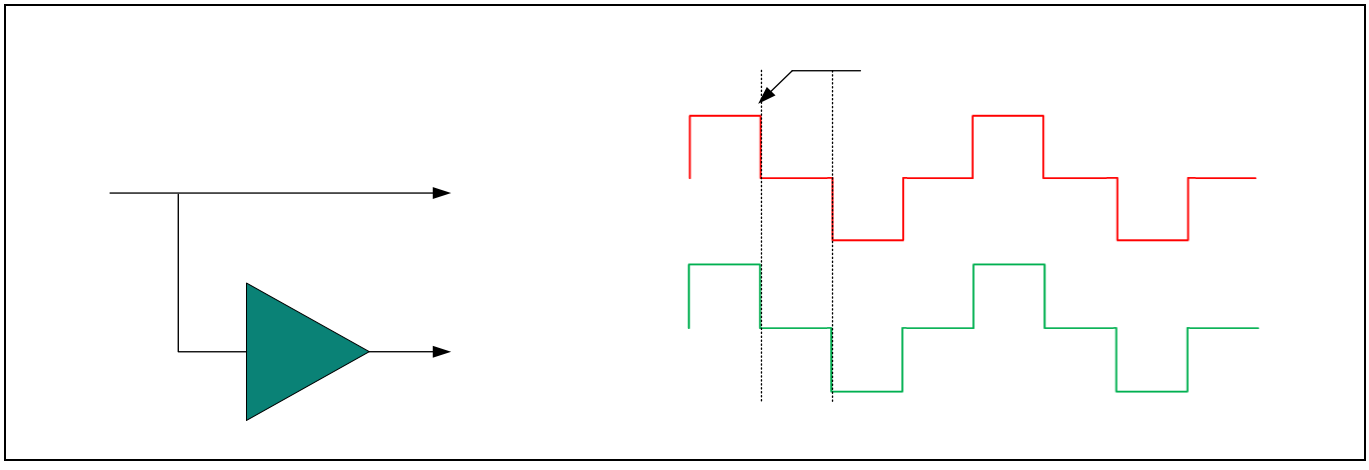


Figure 25 駆動されたシールド信号

Note: シールドを有効にするためには、シールドの静電容量が完全に充電および放電できるように、シールドを駆動するクロック周波数(センスクロック) が十分に低い必要があります。

4.3.2 パッシブシールド

パッシブシールドモードでは、バッファは使用されず、Figure 26 に示すようにシールドが V_{DDA} と GND の間で切り替えられます。切り替えは、センサーとシールド間の正味電荷が 2 センスクロックごとに無効になるように制御されます。

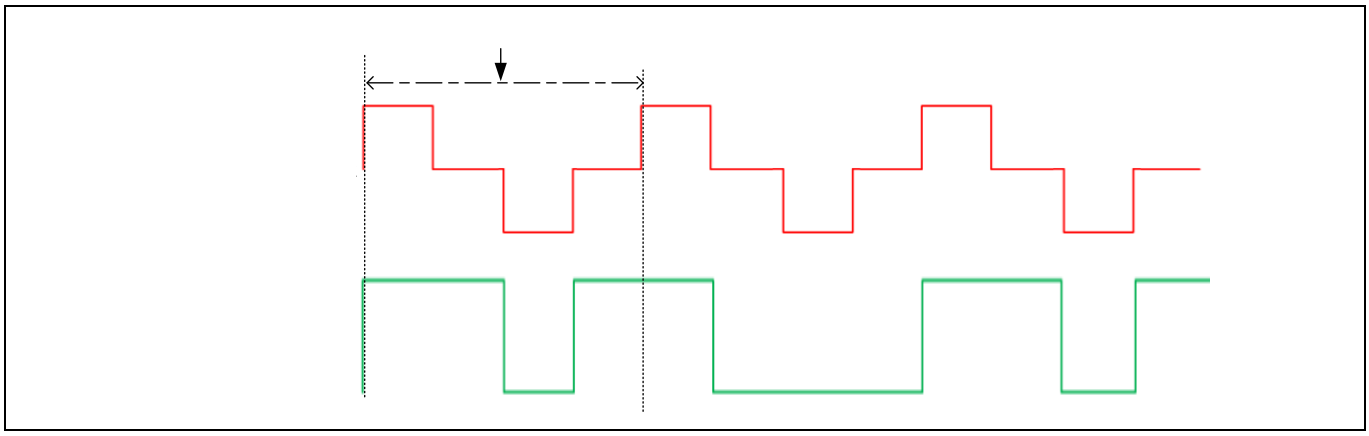


Figure 26 パッシブシールド信号

4.3.3 近接センシング用のシールドの設定

Table 9 に、近接センシング用の設定可能なシールド パラメーターとその推奨値を示します。

Table 9 近接センシング用のシールドパラメーター

パラメーター	説明	推奨値
非アクティブなセンサー接続	システム内の未使用のセンサーへの接続。シールドまたはグランドに設定できます。	このパラメーターを値「シールド」に設定すると、C_P が減少し、耐液体性が実現します。
シールドモード (第 5 世代 CAPSENSE™)	前に説明したように、シールドがアクティブ モードまたはパッシブモードで駆動される方法を定義します。	設計で低消費電力が要求される場合を除き、アクティブシールドモードを推奨します。その場合はパッシブモードを使用できます。
総シールド数	シールドに接続できるセンサー/接続の数を選択します。非アクティブなセンサー (「シールド」に設定されている場合) も、別個のシールドとしてカウントされます。	ほとんどの設計は 1 つの専用シールド接続で機能しますが、一部の設計では、PCB レイアウトの配線を容易にしたり、駆動電流を削減したりするために複数の専用シールド電極が必要になります。
センスクロック分周器	システムクロック (IMO クロック周波数) への分周器。結果として得られるクロックは、センサーとシールドの駆動に使用されます。	32 (IMO クロック周波数 = ~46 MHz の場合)。値が高いほどシールドドライブに適していますが、センサーのスキャン時間も長くなります。

4.4 ウェイクオンアプローチ

ウェイクオンアプローチ (WoA) とは、MCU デバイスが必要な場合にのみアクティブになり、入力を待機しているときにスリープやディープスリープなどの低電力モードに入るスキームを指します。これは、システムのユーザーへの応答性を維持しながら、システムの消費電力を大幅に削減する技術です。

一般的なアプリケーションは、次のいずれかの方法を使用して WoA を実装します。

- ディープスリープを維持し、定期的な割込みソースでウェイクアップします。ウォッチドッグタイマーはそのようなソースの 1 つになり得ます。
- ディープスリープを維持し、GPIO 割込みトリガーなどの外部イベントでウェイクアップします。
- ディープスリープを維持し、上記 2 つのいずれかでウェイクアップします。

第 5 世代 CAPSENSE™を搭載した PSoC™デバイスは、CAPSENSE™ペリフェラルをアクティブに保ち、低電力常時オンセンシング (LP-AoS) と呼ばれるディープスリープモードでセンサーをスキャンする機能を提供します。LP-AoS モードは、ウェイクオンアプローチの実装に効果的に使用できます。ウェイクオンアプローチでは、低電力近接センサーによって物体の接近が検出され、デバイスがウェイクアップし、CPU による通常の動作が可能になります。

タッチベースのセンシングが主なインターフェースであるアプリケーションでは、ユーザーは頻繁にパネルを操作しない可能性があります。システム内のすべてのタッチセンサーを連動させ、非常に低いスキャン周波数で単一の近接センサーとしてスキャンできます。近接が検出されると、デバイスは連動センサーのスキャンを停止し、より高いスキャン周波数で各センサーを個別にスキャンし、指の接触を検出します。このような使用例では、既存のボタンセンサーのセンサーの連動を効果的に使用して、ウェイクオンアプローチ技術を実装できます。

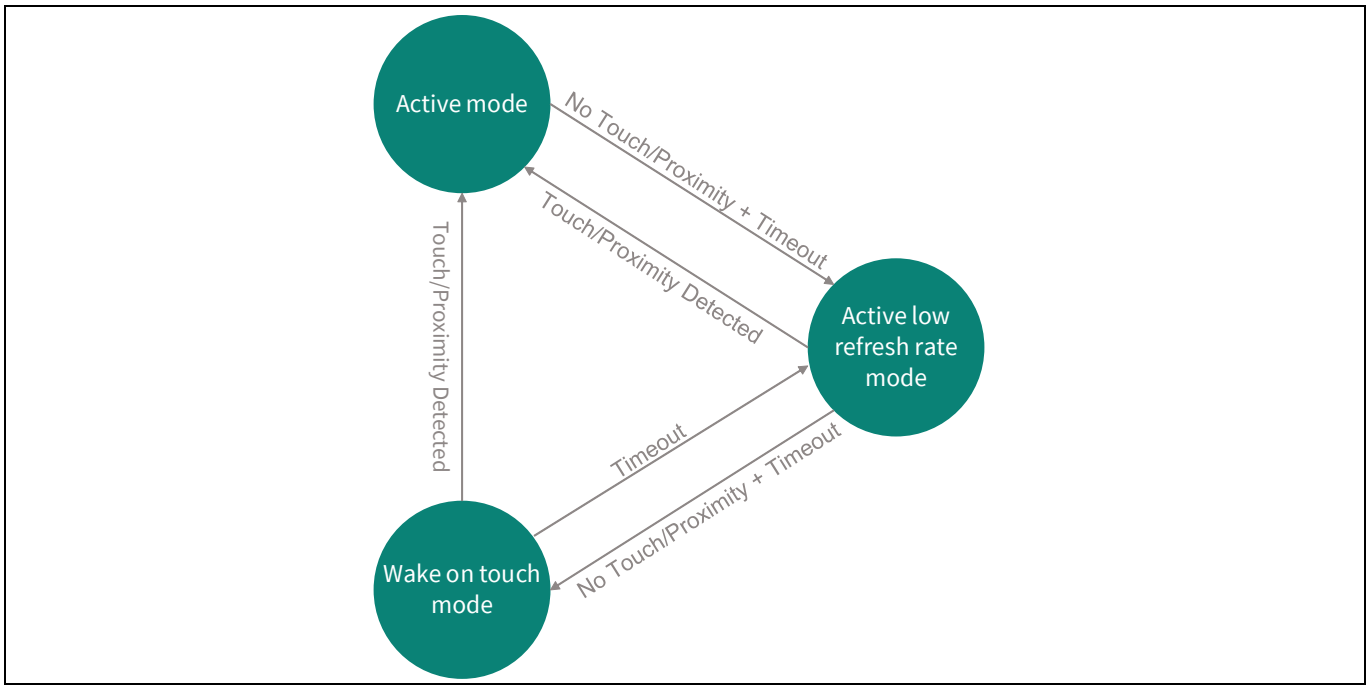


Figure 27 アプリケーションにおけるウェイクオンアプローチの実装例

ワイヤレスコンピュータマウスは、CAPSENSE™近接センシングを使用したウェイクオンアプローチが、電力を節約しながら常時オンの感触を提供する完璧なソリューションとなる例です。マウスは、使用していないときはディープスリープモードに設定でき、手が近づくとすぐにアクティブモードに設定できます。手の接近自体によってシステムが起動されるため、最初のタッチは通常の操作として考慮され、常時オンの感触が得られます。

4.5 センサーの連動

センサーの連動とは、Figure 28 に示すように、複数のタッチセンサーを CAPSENSE™回路に同時に接続し、単一の近接センサーとしてスキャンすることを指します。

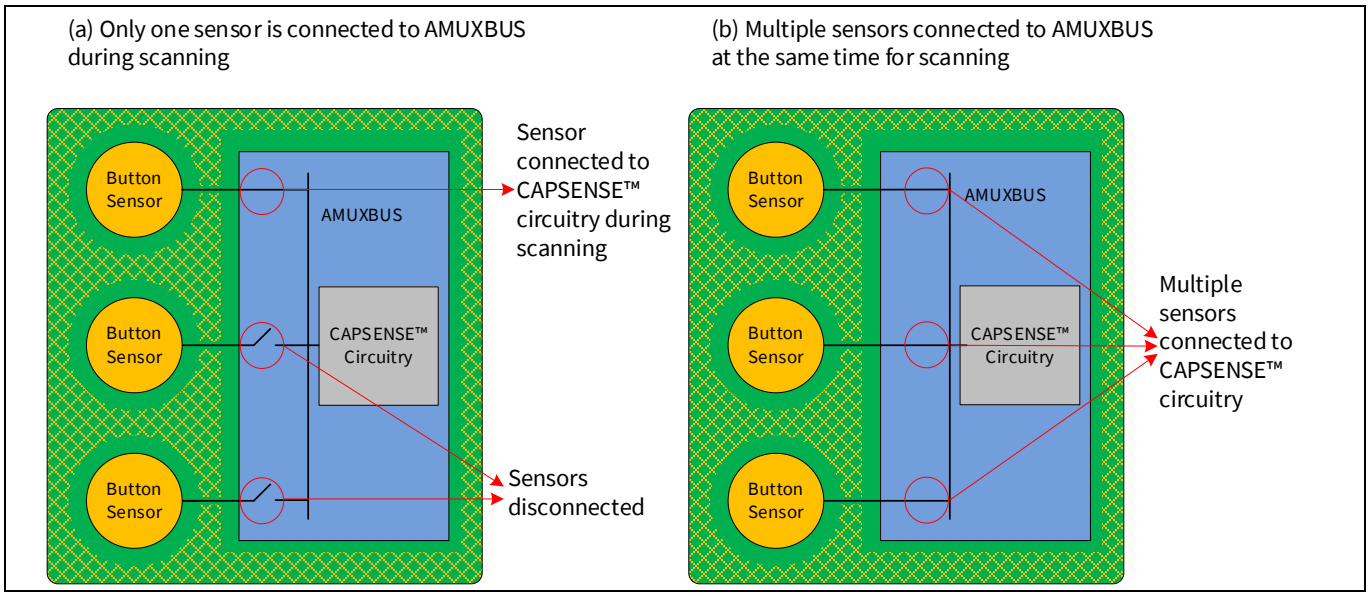


Figure 28 センサー連動による近接センシング

近接センサーは、複数のボタンセンサーまたは近接センサーを連動することで実装できます。複数のセンサーを連動すると、有効センサー面積が増加し、近接検知距離が長くなります。ModusToolbox™の CAPSENSE™コンフィギュレータは、センサーの連動を実装する簡単な方法を提供します。

3 つのボタンセンサーを備えた CAPSENSE™ システムを考えてみましょう。 これらのセンサーを連動するには、次の手順に従います。

1. Figure 29 に示すように、3 つのボタンウィジェットが設定にすでに存在すると仮定すると、CAPSENSE™コンフィギュレータの **Basic** タブに近接ウィジェットを追加し、センシングエレメントの数を 3 に設定してください。

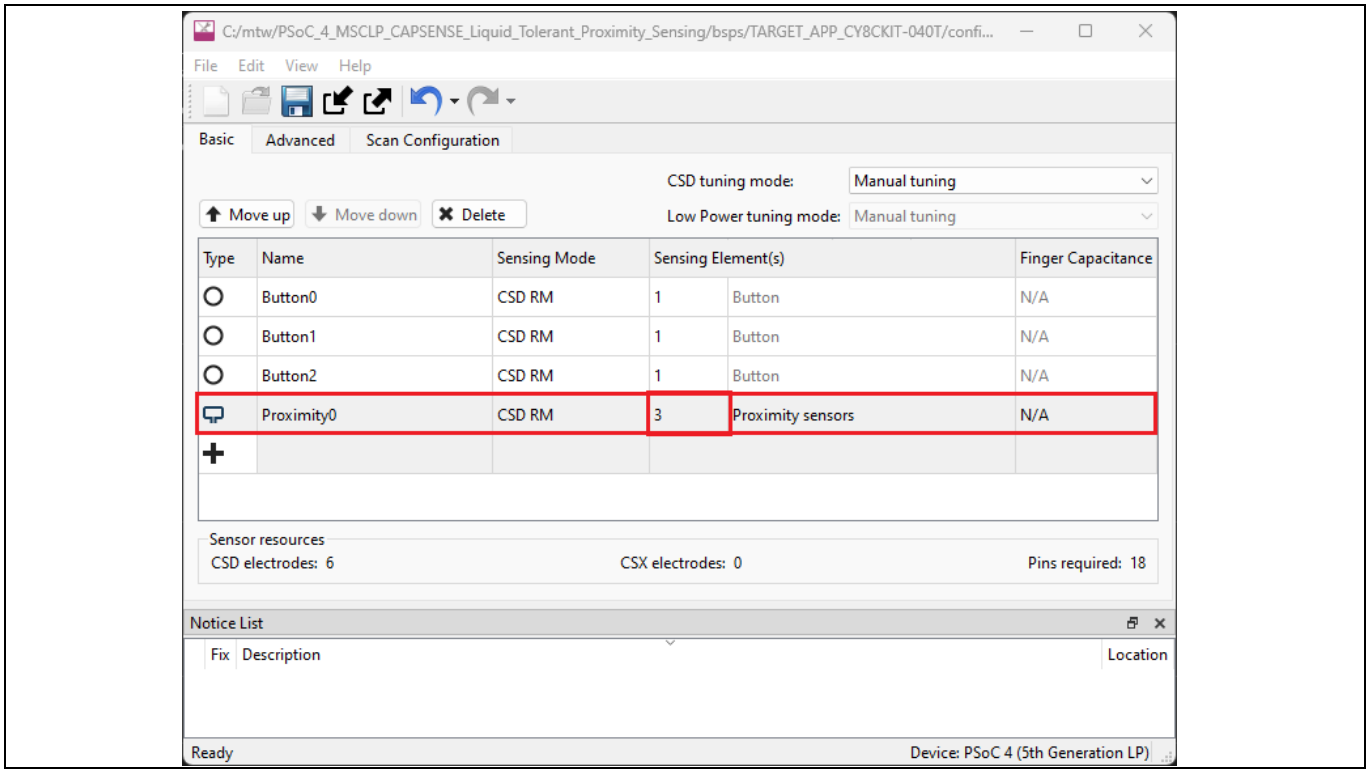


Figure 29 CAPSENSE™ コンフィギュレータでの近接ウィジェットの追加

2. **Scan Configuration** タブに移動し、**Ganged** オプションを選択して、ドロップダウンリストを表示させます。Figure 30 に示すように、すでに設定されている 3 つのボタンセンサーを選択してください。ドロップダウンリストでは、システム内のどのセンサーを単一の近接センサーとして連動させてスキャンするかを指定できます。
3. このウィジェットを近接センサーとして設定し、スキャンできるようになりました。

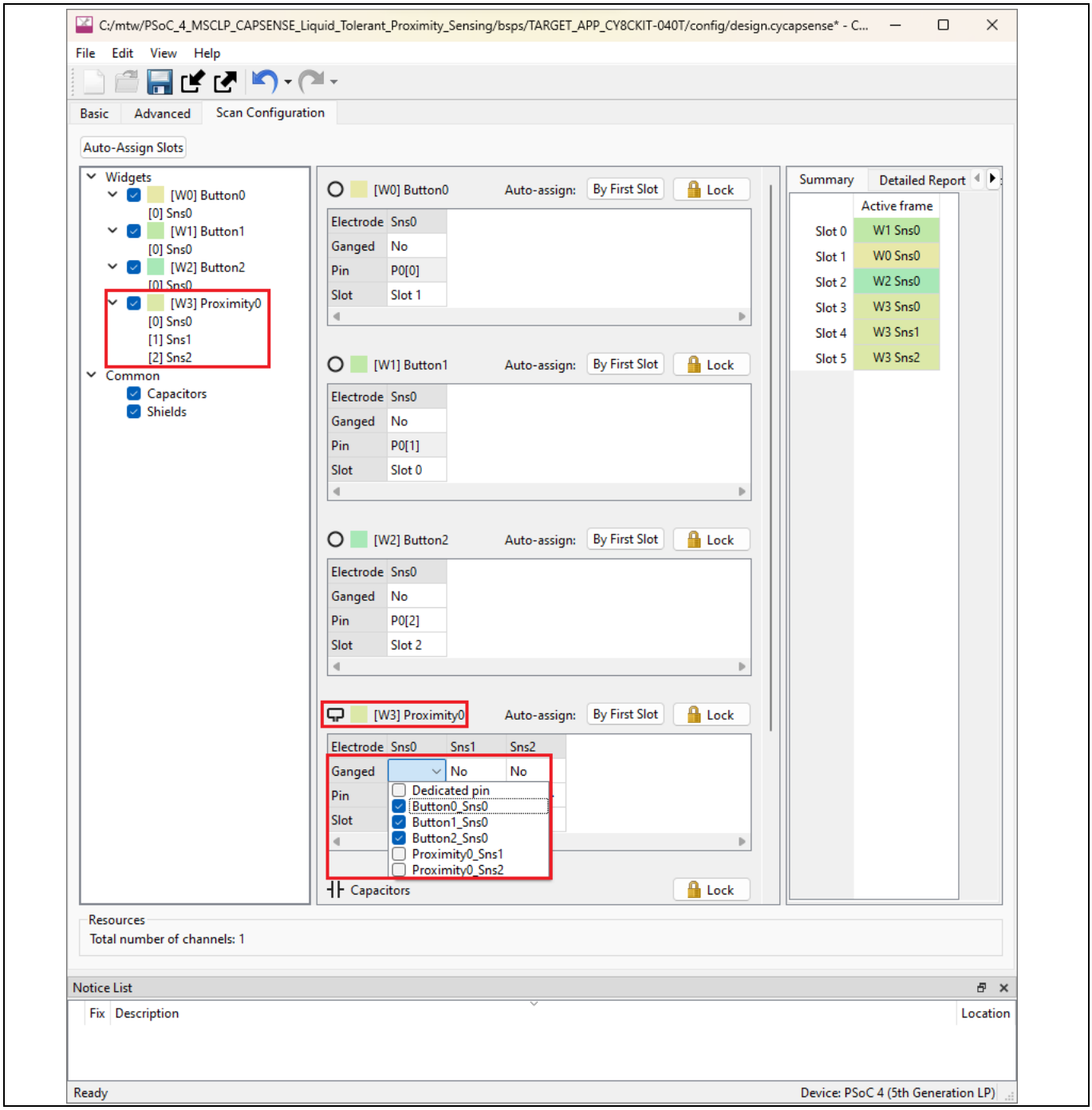


Figure 30 CAPSENSE™ コンフィギュレータの近接ウィジェット上のセンサーの連動

センサーをまとめてスキャンする代わりに専用の近接センサーを使用する場合には、長所と短所があります。

Table 10　連動近接センサーと専用近接センサーの比較

パラメーター	連動近接センサー	専用近接センサー
寄生容量 C_P	高 C_P, より高い解像度, より長いスキャン時間, より高い電力が必要	より低い C_P
チューニング	C_P が高いため、信頼性の高いパフォーマンスを実現するための調整が困難	簡単で信頼性が高い
検出範囲	連動したセンサーに依存し、通常は同等の専用近接センサーよりもはるかに低い	近接ループの直径/対角の 1 ～ 1.5 倍
リソース	追加の MCU ピン, 専用センサー, または専用スペース (PCB 上の領域など) は不要	専用 MCU ピンと専用スペースを使用

近接した物体の検出とは別に、センサー連動のもう 1 つの使用例は、ウェイクオンアプローチを実装することで MCU の消費電力を削減することです。

4.6　ジェスチャ検出ファームウェアのガイドライン

ファームウェアのジェスチャ検出はアプリケーションレベルで設計する必要があります。2 つのセンサーを使用して検出される最も単純なジェスチャはスワイプで、CAPSENSE™タッチスライダーと同様に実装されます。

Figure 31 に示すセンサー配置では、センサー上で手をスワイプすると、Figure 32 に示すような信号が生成されます。

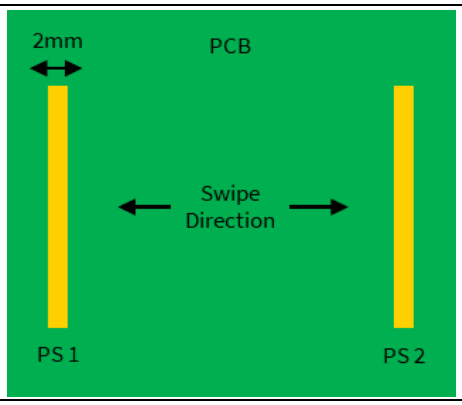


Figure 31　2 つの近接センサーを使用した単軸スワイプ

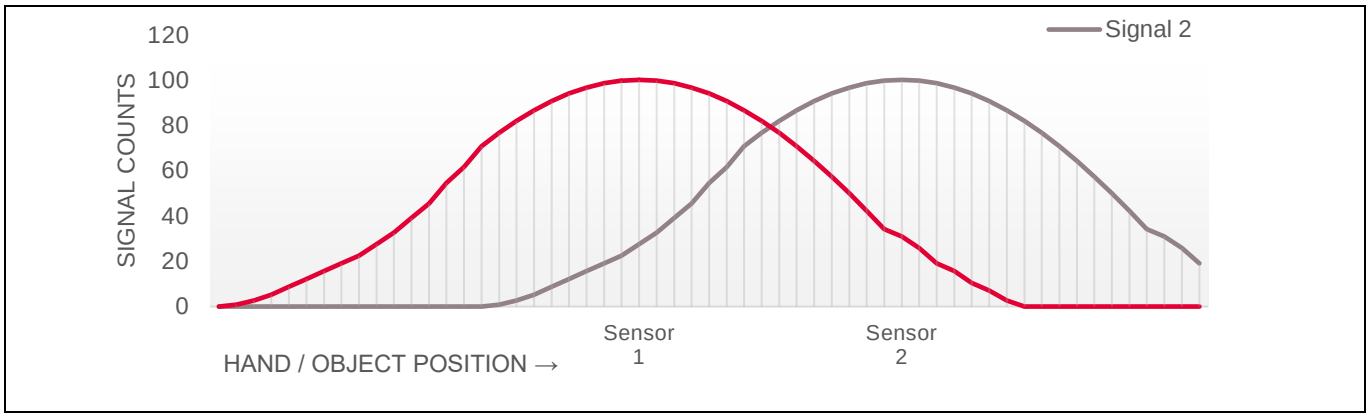


Figure 32　ジェスチャ センサー上のオブジェクト位置の信号カウント

これらの信号を使用してさまざまなジェスチャを検出できます。単純な水平 X 軸スワイプは、これらの信号が定義された制限時間内に次々としきい値を超えるかどうかを追跡するアルゴリズムを使用して検出できます。例えば、X 軸上で左から右へのスワイプが実行される場合、次のように両方のセンサーの差分カウントを 3 つの異なる領域に分割することで手の位置を決定できます。

- **領域 A:** センサーPS1 がオン、PS2 がオフ。手はセンサーPS1 に近く、センサーPS2 からは遠くにあります。
- **領域 B:** PS1 と PS2 の両方がオン。手は PS1 と PS2 の間にあります。
- **領域 C:** PS2 がオン、PS1 がオフ。

手は PS2 に近く、PS1 からは遠いです。左から右へのスワイプが実行されると、最初に PS1 がトリガーされ、次に PS2 がトリガーされます。このトリガーパターンをチェックして、ジェスチャを解釈できます。

近接ジェスチャ検出システムでは、次の属性を使用してジェスチャの方向を測定できます。

- 時間
- 利用可能なすべてのセンサーから (検出されている) 物体までの距離
- トリガーされるセンサーの数

4.7 フィルター

近接センサーはセンサー面積が大きく、感度設定が高いため、ノイズの影響を受けやすくなります。フィルターは raw カウントのノイズを大幅に低減するのに役立ち、SNR と応答時間が向上します。SNR が高いということは、近接検知距離が長く、センシングの信頼性が高いことを意味します。

CAPSENSE™エコシステムには、次のタイプのフィルターがあります。

- ハードウェアフィルター
- ソフトウェアフィルター

これらのフィルターは、何に適用されるかに基づいて、さらに次のように分類されます。

- Raw カウントフィルター
- ベースラインフィルター

ベースラインは信号を計算するための基準として使用されるため、raw カウントよりも安定している必要があります。したがって、raw カウントとは別のフィルターをベースラインに適用する必要があります。

さらに、CAPSENSE™ウィジェットはアクティブまたは低電力として設定できるため、エコシステムはこれらのタイプのウィジェットに対して個別のフィルターを提供します。

Table 11 に、CAPSENSE™エコシステムによって提供されるフィルターの概要を示します。

Table 11 CAPSENSE™ が提供するフィルター

実装	フィルターアプリケーション	フィルタータイプ	アクティブ ウィジェット	低電力 ウィジェット
ハードウェア	Raw カウントフィルター	CIC2	✓	✓
		HW-IIR	✓	✓
ソフトウェア	Raw カウントフィルター	SW-IIR	✓	
		Average	✓	
		Median	✓	
	ベースラインフィルター	SW-IIR	✓	
		SW-IIR fast		✓
		SW-IIR slow		✓

次のセクションでは、近接センシングアプリケーションでこれらのフィルターを効果的に使用および設定する方法について説明します。

4.7.1 ハードウェアフィルター

第 5 世代 CAPSENSE™アーキテクチャを備えた PSoC™デバイスは、次の 2 種類のハードウェア Raw カウントフィルターを備えています。

1. カスケード積分コム 2 (CIC2) フィルター
2. 1 次ハードウェア IIR フィルター

ここでは、これらのフィルターとその設定について詳しく説明します。

4.7.1.1 カスケード積分コム 2 (CIC2) フィルター

デルタ-シグマコンバータ用の 2 次デジタルローパス (デシメーション) フィルターです。これにより、より高い解像度が提供され、それによって特定のスキャン期間の SNR が向上します。

CIC2 フィルターは、デルタ-シグマコンバータである CAPSENSE™アナログフロントエンドの出力を受け取ります。Figure 33 に示すこのフィルターは、移動平均ローパスフィルターと、デシメータとしても知られるダウンサンプラーを組み合わせたものです。IIR フィルターと併用すると、SNR が大幅に向上します。

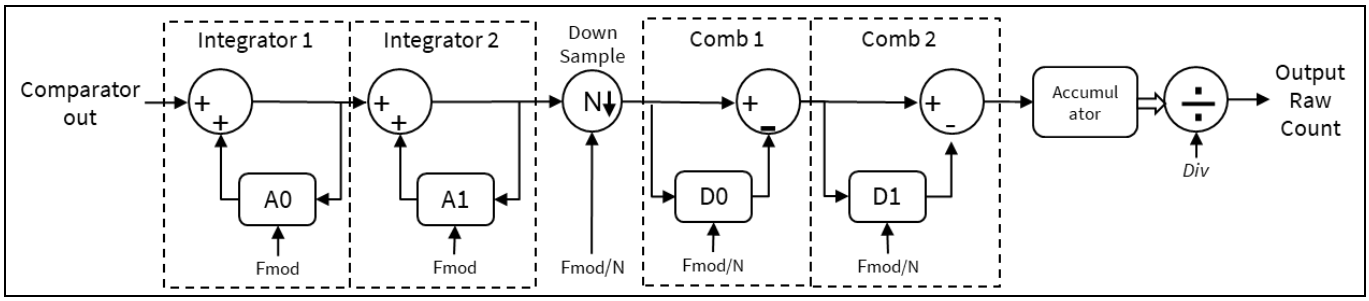


Figure 33 CIC2 フィルターのブロック図

CIC2 フィルターの有効化

Figure 34 に、ModusToolbox™の CAPSENSE™コンフィギュレータの **Advanced** タブで CIC2 フィルターを有効にする方法を示します。

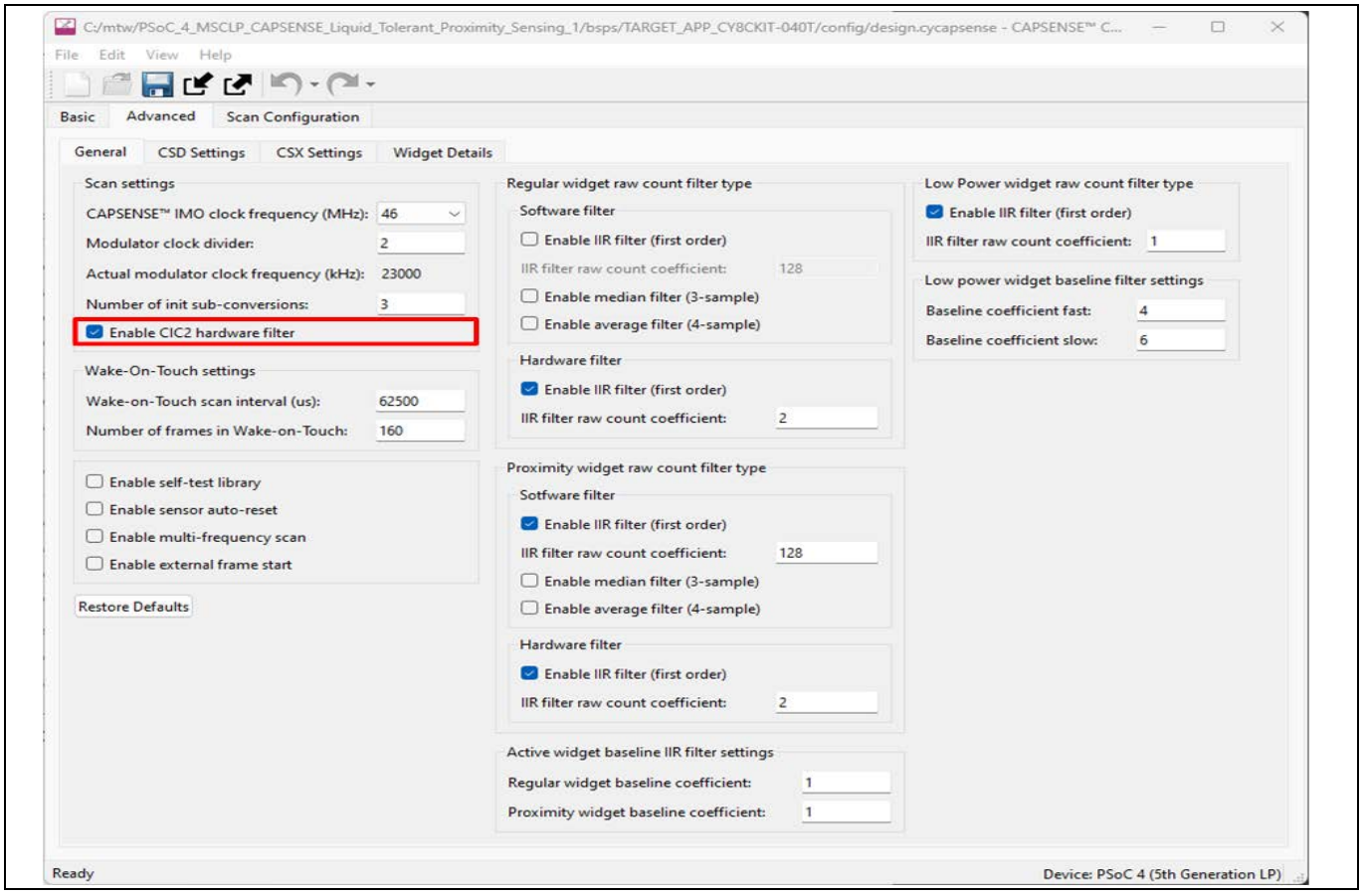


Figure 34 ModusToolbox™の CAPSENSE™コンフィギュレータの Advanced タブでの CIC2 フィルターの有効化

CIC2 フィルターの調整

CIC2 フィルターを調整する目的は、応答時間を最適に保ちながら変換分解能を向上させることです。得られる解像度の向上は、raw カウントの増加に基づいて行われます。センサー容量 (Cs) が一定の場合、raw カウントが高いほど実効解像度が高いことを意味します。

Equation 1 に、達成可能な最大 CIC2 出力 raw カウントを示します。

$$RawCount_{max} = Decimation\ rate^2$$

Equation 1　最大 CIC2 出力 raw カウントの式

効果的な CIC2 フィルタリングには、少なくとも 2 つの有効なサンプルが必要です。2 つの有効なサンプルを考慮すると、最適なデシメーション率は Equation 2 を使用して次のように計算されます。

$$Decimation\ rate\ (N) = \frac{Sns_Clk_Div * Nsub}{3}$$

Equation 2　CIC2 推奨デシメーション率式

Equation 1 からわかるように、デシメーション率が高くなるほど、達成できる最大 raw カウントも大きくなり、その結果、信号の解像度 (ビット数) も高くなります。最適な解像度を実現するには、デシメーション率が Equation 2 に従っている必要があります。

調整するパラメーター

Table 12 に、CIC2 フィルター用に調整できるパラメーターを示します。

Table 12　調整するパラメーター

パラメーター	説明	raw カウント解像度への影響	依存性/トレードオフ
センスクロック分周器	システムクロック (IMO) への分周器。結果として生じるクロックは、センサー信号を駆動するために使用されます。	正比例します	センスクロック分周器の計算はセンサー C_P に大きく依存します。C_P が高くなると必要な周波数が低くなるため、クロック分周器の値も高くなります。 センスクロックが低いとスキャン時間が長くなり、消費電力が高くなることに注意してください。
N_{SUB}	サブコンバージョンの数。 詳細については、近接センシング設計の調整を参照してください。	正比例します	N_{SUB} を増やすとスキャン時間が長くなります。

センス クロック分周器と NSUB の計算の詳細については、AN85951 – PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™デザインガイドを参照してください。

Figure 35 に、ModusToolbox™の CAPSENSE™コンフィギュレータの **Widget Details** タブで調整できる CIC2 フィルターパラメーターを示します。

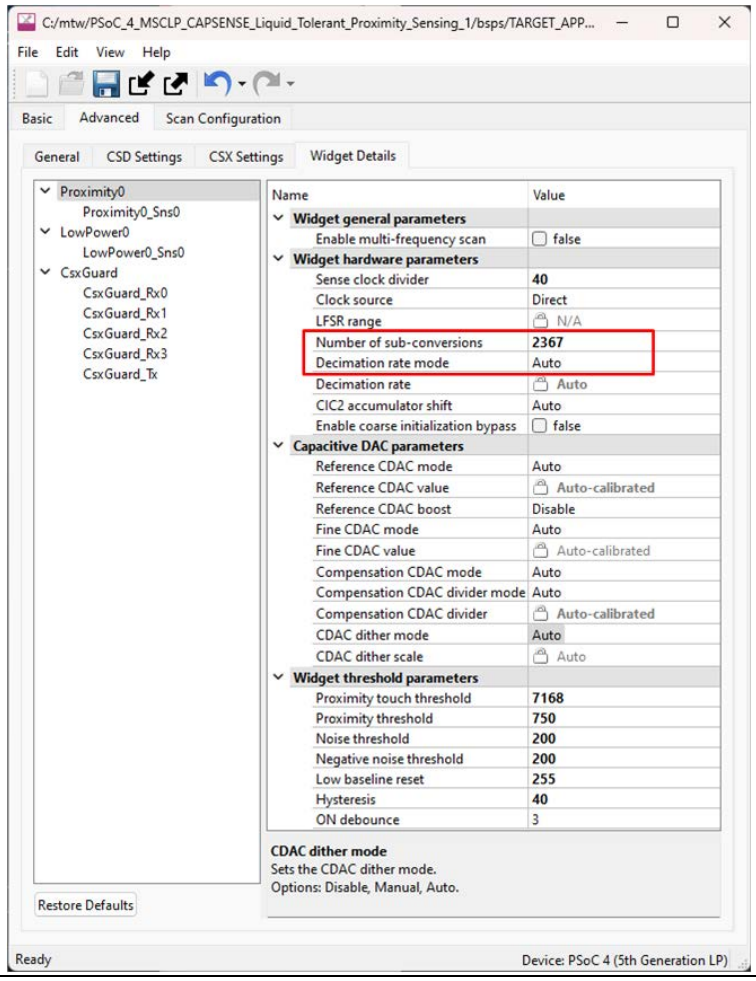


Figure 35 ModusToolbox™の CAPSENSE™コンフィギュレータの Widget Details での CIC2 フィルターの調整

4.7.1.2 一次ハードウェア IIR フィルター

このフィルターは、センサー (アクティブウィジェットと低電力ウィジェットの両方) の raw カウントをフィルターするために使用されます。このフィルターへの入力は次のとおりです。

- CIC2 が有効な場合の CIC2 フィルターの出力
- CIC2 フィルターが無効な場合の CAPSENSE™アナログフロントエンドの出力

Equation 3 に、フィルターの瞬間出力 raw カウントを表します。

$$RawCount = \frac{1}{2^{iirRCcoef}} RawCount_{Current} + \left(1 - \frac{1}{2^{iirRCcoef}}\right) RawCount_{Previous}$$

Equation 3 ハードウェア IIR フィルターの raw カウントの式

ここで、

- **iirRCcoef** – IIR フィルターの raw カウント係数。有効な範囲は 1～8 です。係数が低いとフィルタリングは低くなりますが、応答時間は速くなります。

ハードウェア IIR フィルターの有効化

Figure 36 に、ModusToolbox™の CAPSENSE™コンフィギュレータの **Advanced** タブでハードウェア IIR フィルターを有効にする方法を示します。

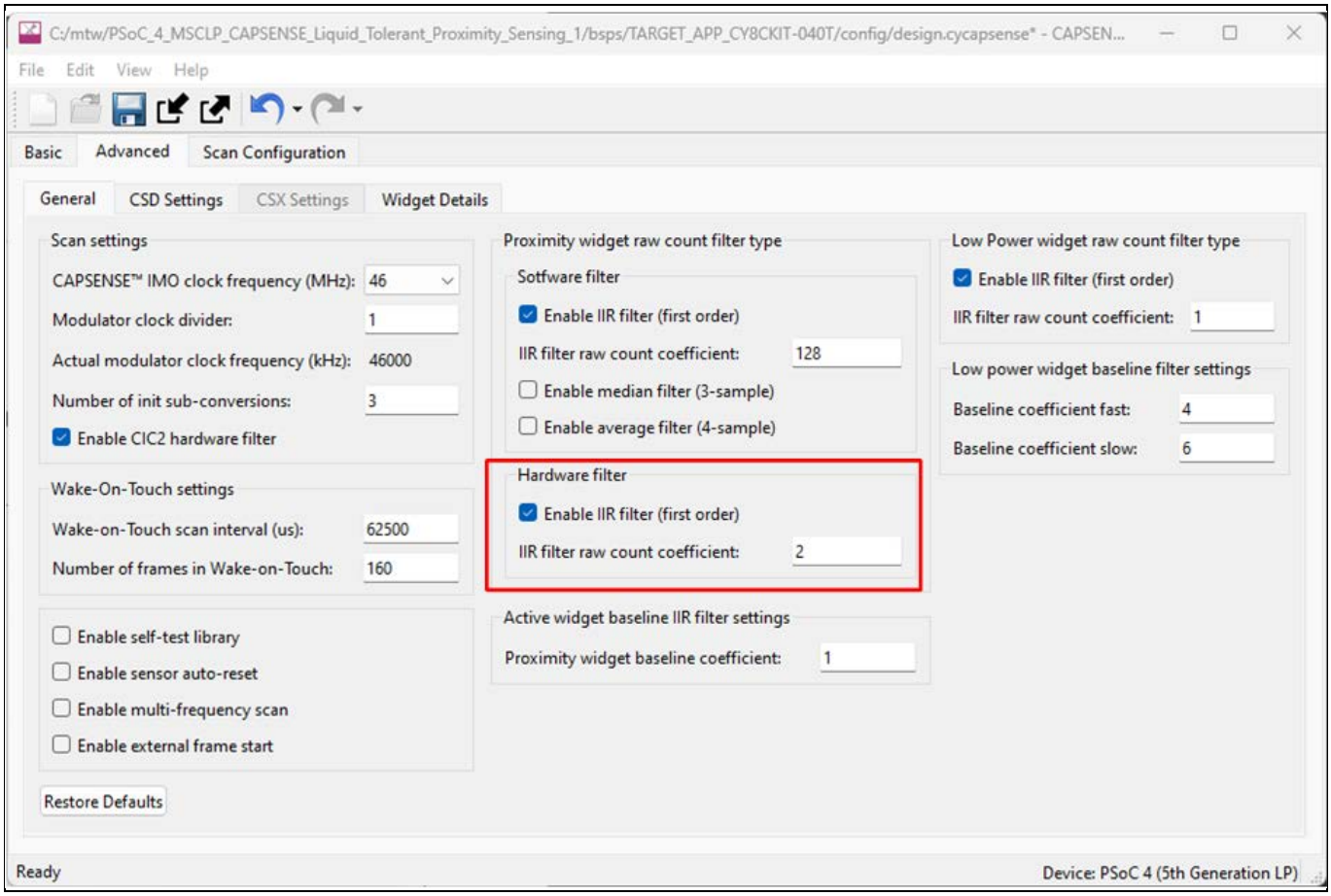


Figure 36 ModusToolbox™の CAPSENSE™コンフィギュレータの Advanced タブでのハードウェア IIR フィルターの有効化

4.7.2 ソフトウェアフィルター

CAPSENSE™エコシステムのミドルウェアライブラリは、以下のソフトウェアフィルターを提供します。

- Raw カウントフィルター
  - 平均
  - 一次 IIR
  - 中央値
- ベースラインフィルター
  - アクティブなウィジェットの一次 IIR
  - 低電力ウィジェット用の低速ベースライン IIR フィルター
  - 低電力ウィジェット用の高速ベースライン IIR フィルター

ソフトウェアフィルターの詳細については、CAPSENSE™入門の「ソフトウェアフィルター」セクションを参照してください。

ソフトウェア IIR フィルター

CAPSENSE™ミドルウェアによって提供されるすべてのソフトウェア IIR フィルターは、次の式で実装されます。

$$Output = \left(\frac{Coefficient}{256} \times input\right) + \left(\frac{256 - Coefficient}{256} \times previous\ output\right)$$

Equation 4　ソフトウェア IIR フィルターの式

ソフトウェアフィルターをプロジェクトに追加

必要なソフトウェアフィルターは、Figure 37 に示すように、CAPSENSE™コンフィギュレータの **Advanced** タブから ModusToolbox™プロジェクトに追加できます。

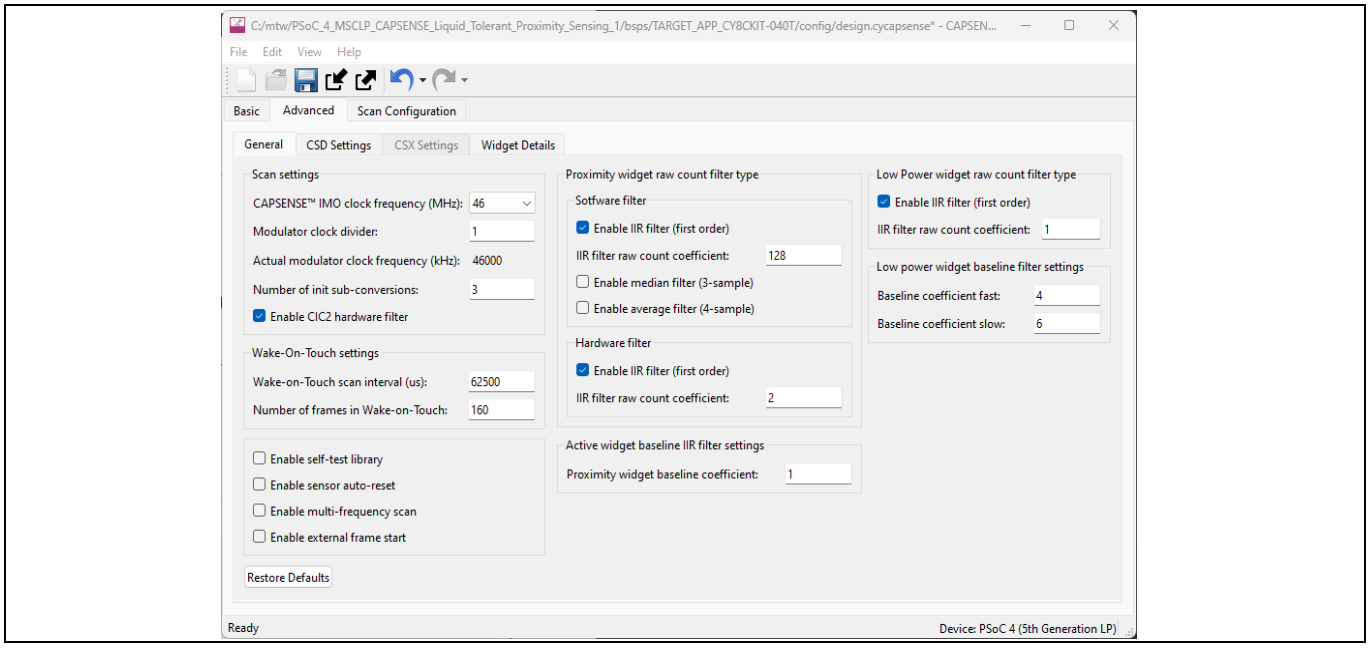


Figure 37　プロジェクトへのソフトウェアフィルターの追加

4.7.2.1　ソフトウェアの raw カウントフィルター

3 種類の raw カウントフィルターが CAPSENSE™ミドルウェアによって提供されており、CAPSENSE™コンフィギュレータツールを使用して設定できます。Table 13 では、これらのフィルター, そのアプリケーション, およびそれぞれの調整可能なパラメーターについて説明します。

Table 13　CAPSENSE™が提供するソフトウェア raw カウントフィルター

種類	説明	アプリケーション	調整パラメーター
Average	Figure 37 に示すように、4 サンプル長の単純な移動平均フィルター	電源からの周期的なノイズ	なし
IIR	単純な RC フィルターと同様に、Figure 37 に示すように、調整可能な IIR 係数を備えたソフトウェア IIR フィルター	高周波ホワイトノイズ (1/f ノイズ)	IIR 係数 – 係数を低くすると、フィルタリング効果が高くなります。有効範囲: 1 ～ 128

種類	説明	アプリケーション	調整パラメーター
Median	移動平均フィルターと同様に、サイズ 3 のバッファから入力値の中央値を計算する非線形フィルター	モーターやスイッチング電源からのノイズスパイク	なし

4.7.2.2 ソフトウェアベースラインフィルター

ベースラインは信号を計算するための基準として使用されるため、raw カウントよりも安定している必要があります。したがって、CAPSENSE™ミドルウェアでは個別のフィルターが提供されており、CAPSENSE™コンフィギュレータツールを使用して設定できます。

Table 14 CAPSENSE™ が提供するソフトウェア ベースライン フィルター

種類	説明	アプリケーション	調整パラメーター
アクティブなウィジェットの一次 IIR	調整可能な IIR 係数を備えたソフトウェア IIR フィルター	高周波ノイズ	IIR 係数 – 係数を低くするとフィルタリング効果が高くなります。有効範囲: 1〜255
低速ベースラインフィルター (第 5 世代 CAPSENSE™)	低速で変化するベースラインの IIR フィルター係数 (低電力ウィジェットのみ)	高周波ホワイトノイズ	IIR 係数 – 係数を低くするとフィルタリング効果が高くなります。有効範囲: 1〜15
高速ベースラインフィルター (第 5 世代 CAPSENSE™)	高速で変化するベースラインの IIR フィルター係数 (低電力ウィジェットのみ)	高周波ホワイトノイズ	

4.8 耐液性に関するファームウェアのガイドライン

ここでは、耐液ファームウェア設計のガイドラインについて説明します。

4.8.1 自己容量の使用

CAPSENSE™設計の耐液性 (防沫性) を実現する簡単な方法は、感知面に液体の飛沫が存在するときに検出しきい値を受信する raw カウントの 3 倍に設定することです。さらに、液体の存在による信号を最小限に抑えるために、耐液性に関するレイアウトのガイドラインで説明されているシールドの実装が推奨されます。

耐液性を考慮した CAPSENSE™ CSD 設計の調整の詳細については、耐液性のチューニングを参照してください。

設計が近接センサーのみで構成され、タッチ検出が必要ない場合、Figure 38 に示すように、大量の液体の検出に最大制限しきい値を使用できます。大量の液体の存在 (またはセンサーを液体に浸した場合) の raw カウントは、近くにあるターゲットオブジェクトの raw カウントよりも大幅に高くなります。液体の存在を検出するためのしきい値は、近接して検出される対象物体がセンサーに最も近いときに達成できる最大 raw カウントよりも高くなければいけません。この手法はアプリケーションに実装する必要があることに注意してください。

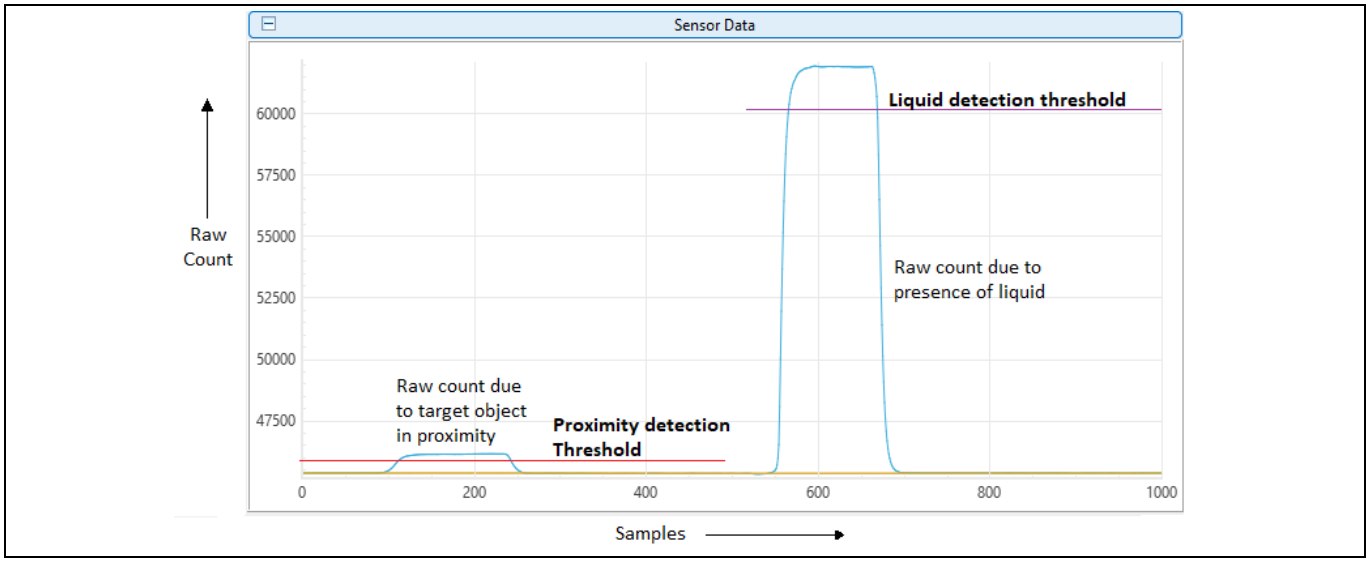


Figure 38　近接検出信号と比較した液体検出のしきい値

4.8.2　CSX ガード センサーの使用

CAPSENSE™ベースの近接センシングにおける耐液体性の推奨手法の 1 つは、耐液性に関するレイアウトのガイドラインで説明されているように、相互静電容量 (CSX) ガードセンサーを使用することです。

この技術は、はじめにで説明したように、CSX ベースのセンサー上に液体が存在すると、Tx 電極と Rx 電極間の相互静電容量が増加するという事実に基づいています。ただし、人間の手 (または接地された物体) の存在により、相互静電容量 (C_M) が減少します。

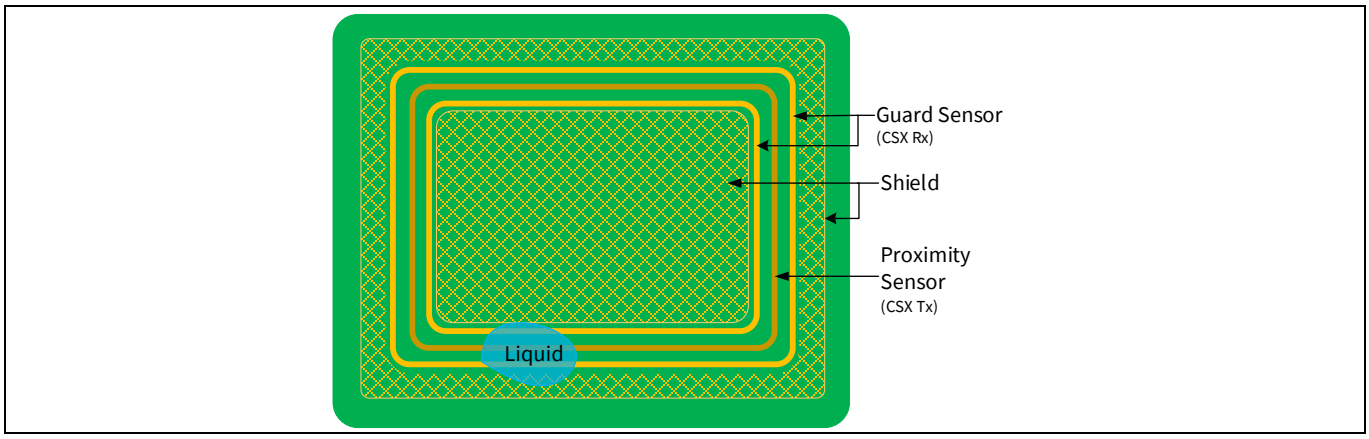


Figure 39　ガードセンサーの PCB レイアウト

Figure 39 に示すように、センサー設定に液体が存在するシナリオを考えてみましょう。Figure 40 に示す設定とアルゴリズムをアプリケーションに実装して、耐液性を実現できます。

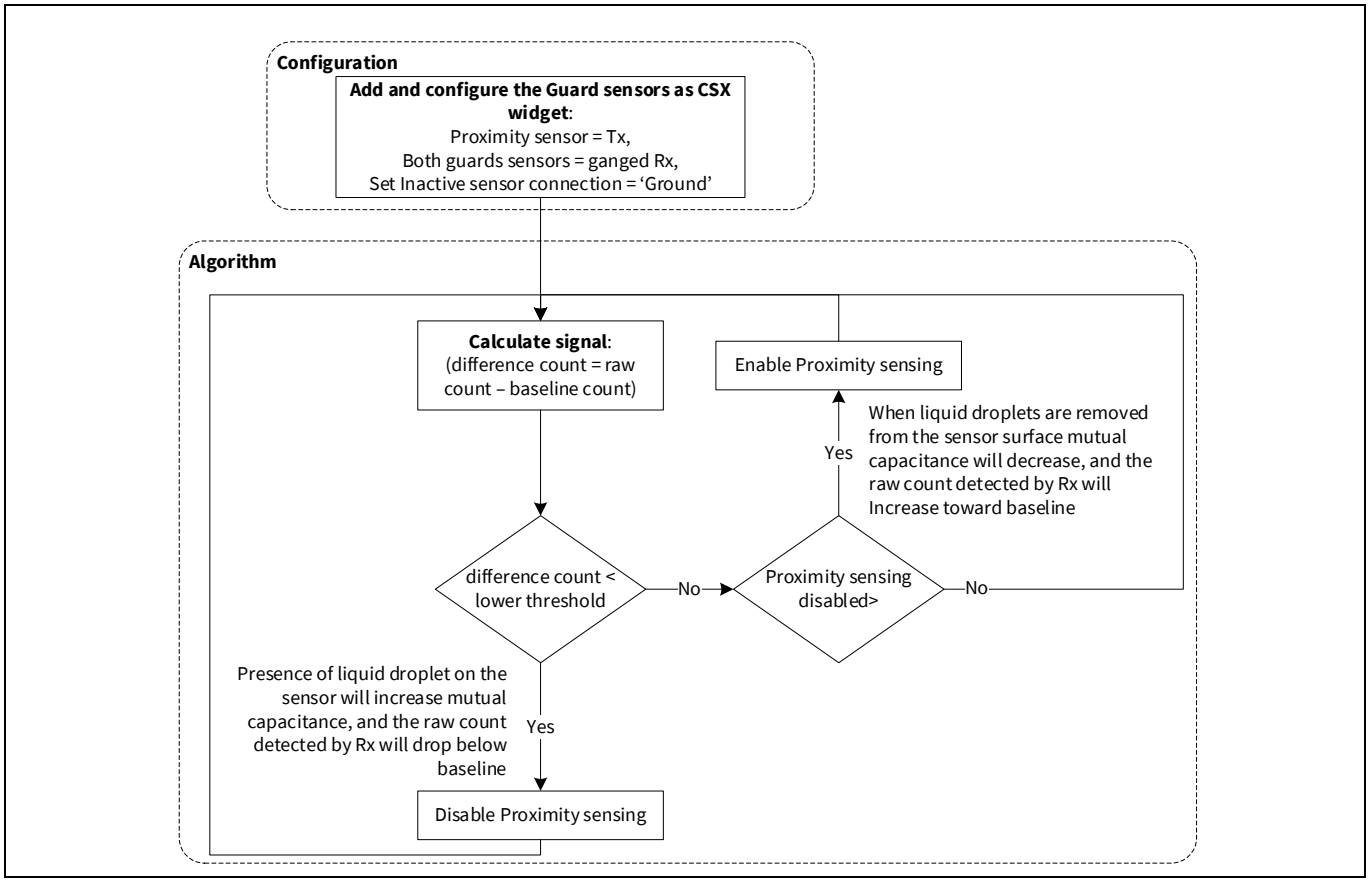


Figure 40　CSX ガードセンサーの設定とアルゴリズム

この手法は、ミドルウェア API を使ってアプリケーションに実装する必要があることに注意してください。詳細については、PSoC™ 4: CAPSENSE™ MSCLP Liquid tolerant proximity sensing を参照してください。

ここで説明する方法を組み合わせて使用すると、堅牢な耐液性を実現できます。

4.9 ミドルウェアライブラリと API

インフィニオンは、ここ (ファームウェア設計ガイドライン) で説明されている機能を実装するために使用できる、CAPSENSE™製品およびソリューション用のミドルウェアライブラリを提供します。

Figure 41 に示すように、ModusToolbox™クイックパネルで利用可能な CAPSENSE™ミドルウェアライブラリのヘルプドキュメントを参照してください。

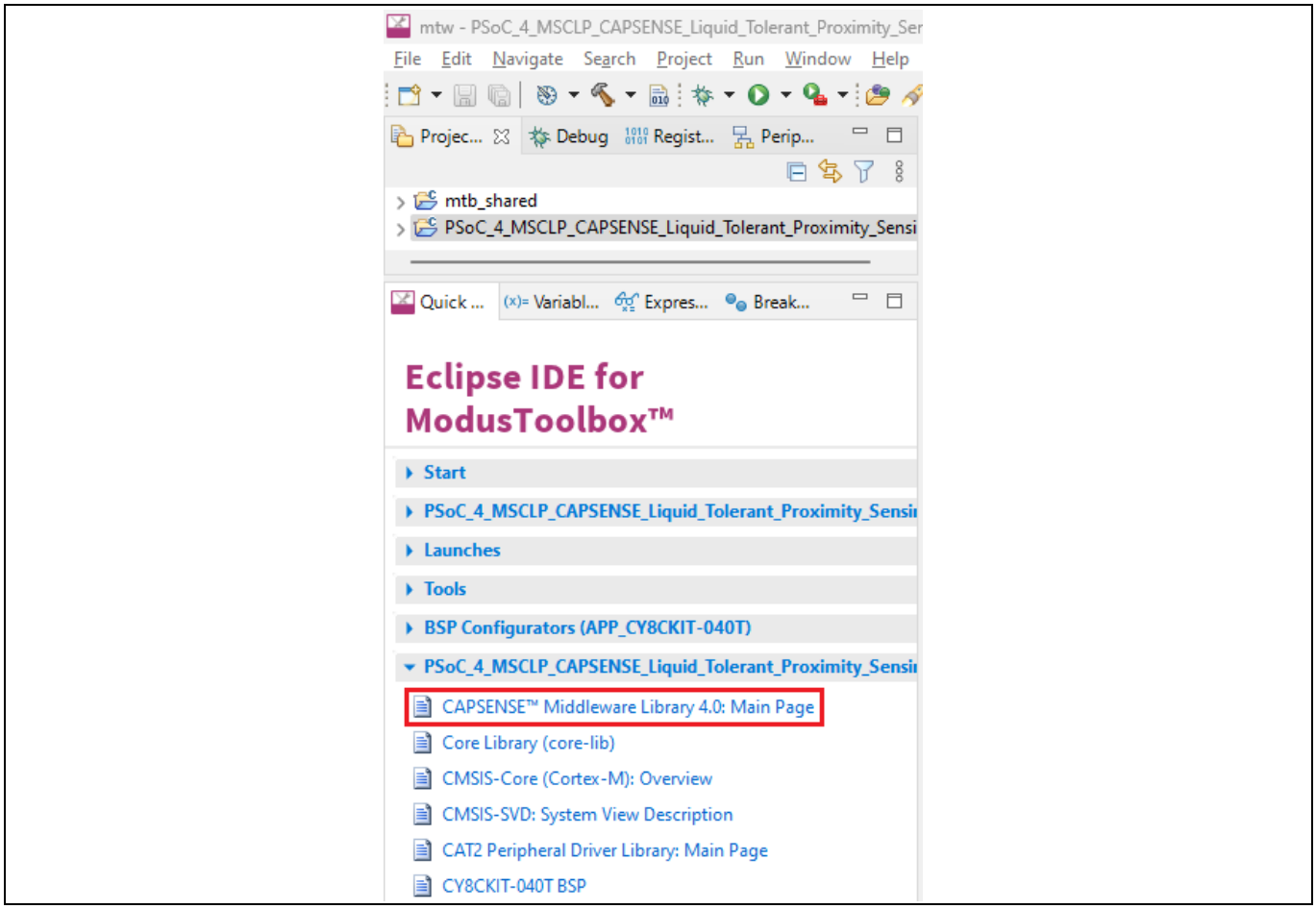


Figure 41 ModusToolbox™内の CAPSENSE™ミドルウェアライブラリのヘルプドキュメント

PSoC™ 4: MSCLP CAPSENSE™ low-power proximity tuning で入手できるサンプルコードは、インフィニオンの CAPSENSE™ミドルウェアライブラリとそれぞれの API の使用方法を学ぶための優れた出発点として役立ちます。

5 チューニングガイドライン

近接センサーのレイアウトとファームウェアの準備ができたら、次のステップは、最適なパフォーマンスを達成するために近接センサーの CAPSENSE™ CSD パラメーターを調整することです。

ボタン センサーに触れた場合の静電容量の追加は数百フェムトファラドとは異なり、近接センサーから離れた対象物によって追加される静電容量は数十フェムトファラドになります。このような小さな静電容量の変化を検出するには、CAPSENSE™回路を高感度に調整し、しきい値パラメーターを最適な値に設定する必要があります。センサーのパフォーマンスを最適化するために CAPSENSE™ CSD パラメーターを設定するプロセスを「チューニング」と呼びます。ここでは、最適なパフォーマンスを達成するために、近接センサーの ModusToolbox™の CAPSENSE™パラメーターを調整する方法について説明します。

5.1 信号対雑音比 (SNR)

最適なセンシングを実現するために静電容量センサーを調整する前に、近接センシングのコンテキストにおける SNR の概念とその計算方法を理解することが重要です。

経験則として、信頼性の高いセンシングには 5:1 以上の SNR が必要です。SNR が高いほど、より高い感度を達成でき、その結果、センシング距離も長くなります。

SNR の計算方法

SNR 測定の最初のステップは、OFF シナリオと ON シナリオで各センサーの raw カウントを監視することです。OFF シナリオは、センサーの近くにターゲット オブジェクトが存在しない場合です。ON シナリオは、対象物体が存在し、センサーによって検出される場合です。SNR を測定するには、それぞれ少なくとも 3000 個のサンプルを記録することを推奨します。

考慮すべきもう 1 つの要素は、信号がどのように生成されるかです。SNR を測定するときは、最悪の場合の ON および OFF のシナリオを使用する必要があります。近くにある人間の手の存在を感知するようにシステムが設計されている場合は、センサーから必要な最も遠い距離にある手の存在で SNR を測定します。

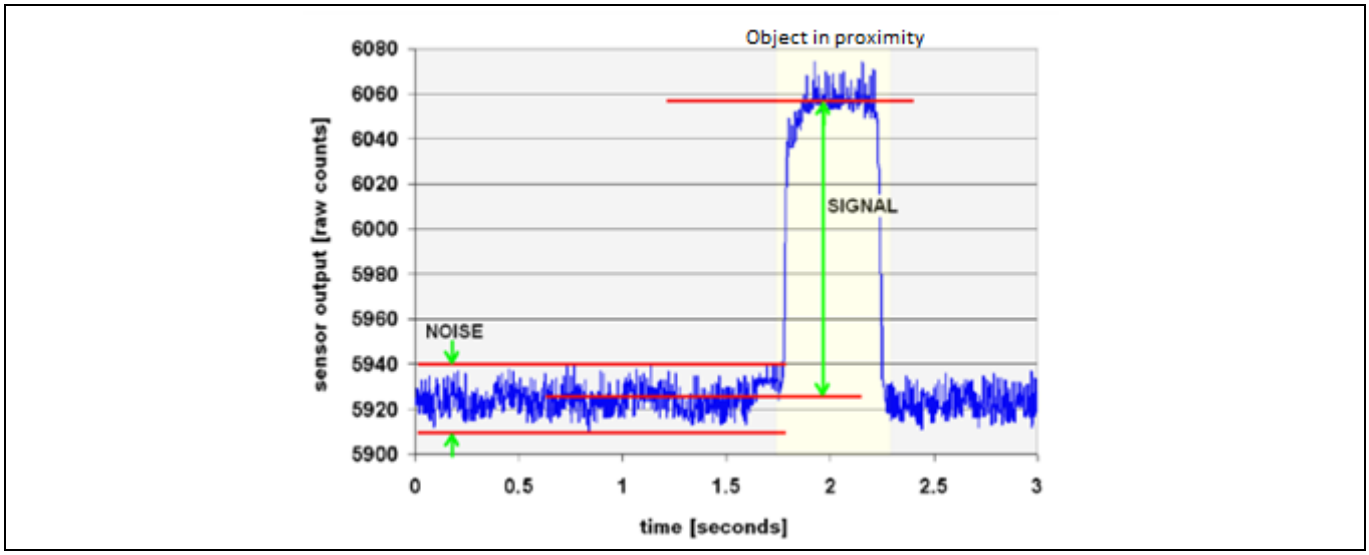


Figure 42 CAPSENSE™信号

SNR の測定例として、Table 15 に示す計算を使用した、Figure 42 の raw カウント波形を考えてみましょう。

Table 15　SNR を計算するためのパラメーター

パラメーター	値
手を近づけない場合の最小 raw カウント	5910 カウント
手を近づけない場合の最大 raw カウント	5940 カウント
手が近くにない場合の平均 raw カウント (ベースライン)	5925 カウント
手を近づけた場合の平均 raw カウント	6055 カウント
差分カウント (信号)	6055 – 5925 = 130 カウント
ノイズカウント	5940 – 5910 = 30 カウント
SNR	**130:30 = 4.3 : 1**

ModusToolbox™を使用した SNR の測定

これを容易にするために、ModusToolbox™は、Figure 43 に示すように、CAPSENSE™設計用のシンプルな SNR 測定ウィザードを提供します。ただし、この方法はセンサーデータを読み出すための MCU との I2C 通信のみをサポートしており、アプリケーションが特定の API セットを実行する必要があります。ファームウェアで。PSoC™ 4 CAPSENSE™ Component datasheet の「Tuner GUI Interface」セクションを参照してください。



Figure 43　ModusToolbox™の CAPSENSE™ Tuner を使用した SNR 測定

Note: SNR は、CAPSENSE™の使用が意図されているノイズ環境で測定する必要があります。言い換えれば、最悪のノイズ条件下でシステムの SNR を測定します。

CAPSENSE™ Tuner を使用して SNR を計算するには、次の手順に従います。

1. ModusToolbox™を実行しているシステムにハードウェアを接続してください。
2. CAPSENSE™ Tuner を開き、**SNR Measurement** タブに切り替えてください。
3. **Widget Explorer** ウィンドウで近接センサーを選択し、**Acquire Noise** をクリックしてください。

4. ノイズを取得した後、対象物をセンサーの近接範囲に持ってきて、**Acquire Signal** をクリックします。信号取得が進行している限り、ターゲットオブジェクトが近接ループの上に留まるようにしてください。

選択したセンサーの計算された SNR がウィンドウに表示されます。

CAPSENSE™ Tuner を使用した SNR 測定の詳細については、PSoC™ 4: MSCLP CAPSENSE™ low-power proximity tuning を参照してください。

Table 16 では、SNR を最大化するために最も関連性の高い CAPSENSE™パラメーターについて説明します。

Table 16　CAPSENSE™調整パラメーターが SNR に及ぼす影響

パラメーター	SNR との関係	トレードオフ
センスクロック分周器	センスクロック分周器は最大 raw カウント (CIC2 フィルターが有効な場合) に直接比例するため、スキャンの解像度が高く、解像度が高いほど SNR が向上します。	センスクロック分周器が高いほど、スキャン時間が長くなります。
サブコンバージョン数 (N_{SUB})	サブコンバージョン (N_{SUB}) の数を増やすと、信号の解像度と SNR が向上します。	N_{SUB} が大きいほどスキャン時間も長くなります。
シールドモード	アクティブモードに設定されたシールドはセンサー信号によって駆動され、SNR の向上に役立ちます。	アクティブおよびパッシブシールドモードでは、消費電力が増加します。
Raw カウントフィルター係数	フィルターの raw カウント係数が高いほど、SNR も高くなります。	係数が高くなると、応答が遅くなり、消費電力が増加し、スキャン結果の処理時間が長くなります。

フィルター設定の再検討

フィルターは、CAPSENSE™設計の SNR を大幅に改善するのに役立ちます。SNR を向上させようとして、上記のすべてのパラメーターを調整しても SNR が向上しない場合は、フィルターで説明したようにフィルター係数の設定を再検討することを推奨します。フィルター係数を増やすと SNR が向上しますが、処理時間と消費電力も増加します。

5.2　近接センシング設計の調整

近接センサーの調整では、次の主な目的に重点を置く必要があります。

- SNR ≥ 5:1 の確保
- 必要なスキャンと応答時間を満たす
- 必要に応じて、耐液性、低電力

CAPSENSE™ 近接設計の調整手順は、次の段階で説明できます。

1. 初期パラメーターを設定する
2. センサーの静電容量を測定し、CDAC ディザ パラメーターを設定します
3. センスクロック周波数の設定
4. 必要な SNR, 電力, およびリフレッシュレートを微調整する
5. しきい値パラメーターを調整する
6. 必要に応じて耐液性のしきい値パラメーターを微調整します

5 チューニングガイドライン

Figure 44 に、近接センサーを調整するための高レベルの手順を示します。

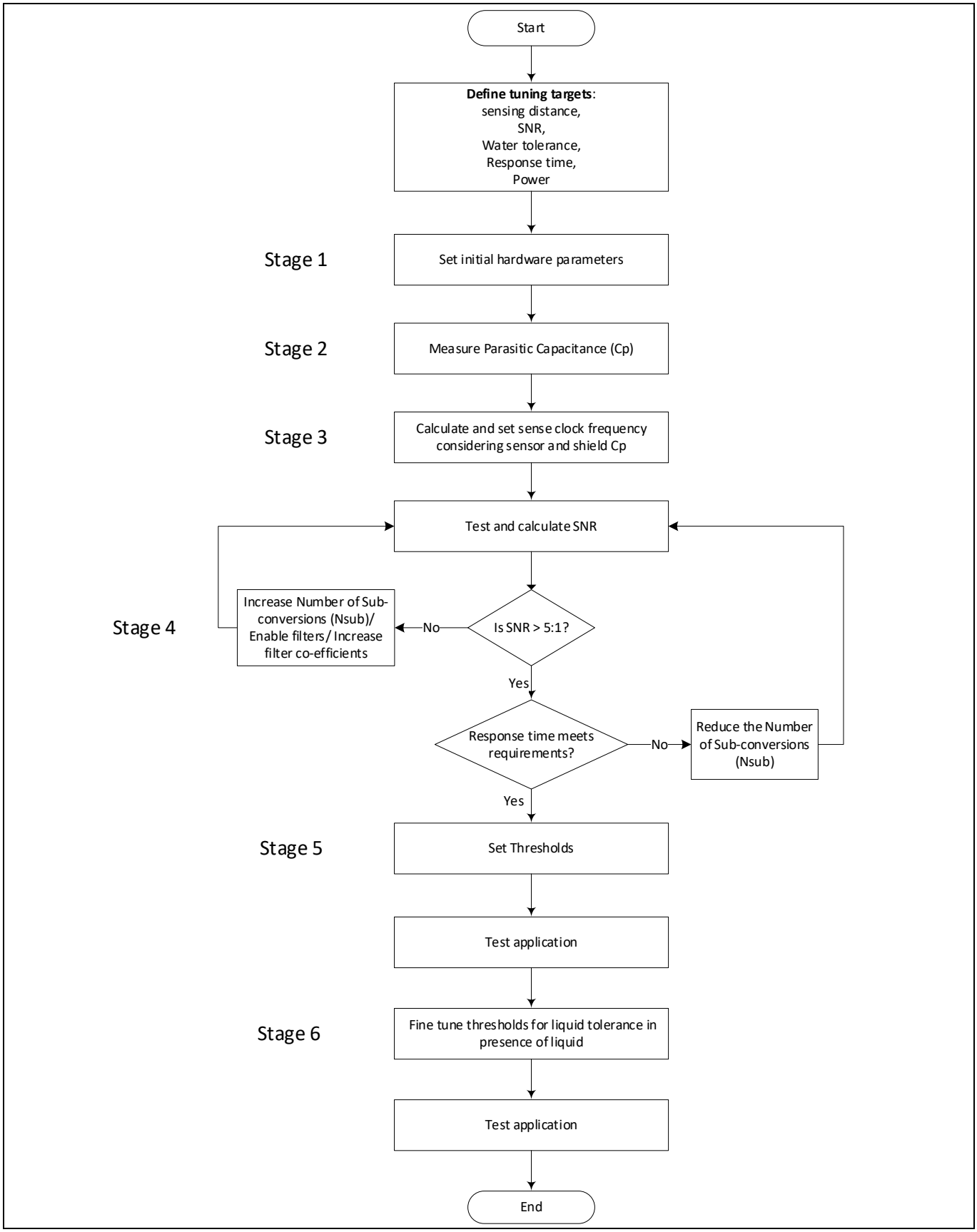


Figure 44　近接センサーを調整するための高レベルの手順

5 チューニングガイドライン

調整するパラメーター

CAPSENSE™エコシステムは、さまざまなタイプの CAPSENSE™タッチおよび近接センシングアプリケーションをサポートするための優れた柔軟性を提供します。Table 17 に、ModusToolbox™で利用可能な、近接センシングに重要なパラメーターとその推奨値を示します。

Table 17　CAPSENSE™チューニングパラメーター

パラメーター	説明	推奨値
IMO クロック周波数	CAPSENSE™ペリフェラルのソースとして使用されるクロックの周波数	デフォルト、つまり 46 MHz (PSoC™ 4000T) のままにしておきます。
変調器クロック分周器	システムクロックを分周するために使用される分周値。結果として得られるクロックは、容量からデジタル変調へのクロックとして使用されます。	シールドが実装されている場合は 2、それ以外の場合は 1
センスクロック分周器	システムクロック (IMO クロック周波数) への分周器。結果として得られるクロックは、センサーとシールドの駆動に使用されます (値は 4 の倍数である必要があります)。	28 (IMO クロック周波数 = ~46 MHz の場合)
クロックソース	CAPSENSE™に使用されるクロックのソース。利用可能なソースは 3 つあります。 1. 直接 2. 疑似ランダムシーケンス (PRSx) 3. スペクトラム拡散クロック (SSCx)	デフォルト、つまり「直接」のままにしておきます。
サブコンバージョン数 (N_{SUB})	サブコンバージョン数によって、センサーの感度とセンサーのスキャン時間が決まります。固定の変調器クロックとセンスクロックの場合、サブ変換 (N_{SUB}) の数を増やすと、信号の分解能と SNR が向上します。ただし、サブコンバージョン数を増やすと、次の方程式に従ってセンサーのスキャン時間も増加します。 $Scan\ time = \frac{N_{SUB}}{\text{Sense clock frequency}}$	応答時間が遅く感じられなくなるまでこのパラメーターを最大化します。 推奨値は 2000 (第 5 世代 CAPSENSE™) です。
デシメーション率 (CIC2) (第 5 世代 CAPSENSE™)	詳細については、CIC2 フィルターのセクションを参照してください。	詳細については、CIC2 フィルターのセクションを参照してください。
CIC2 アキュムレータシフト (第 5 世代 CAPSENSE™)	詳細については、CIC2 フィルターのセクションを参照してください。	詳細については、CIC2 フィルターのセクションを参照してください。
近接しきい値	近接で検出されたオブジェクトに対してアルゴリズムが正の結果を返す raw カウント	センサーから必要な最大距離にある信号の 80%
タッチしきい値	センサーで検出されたタッチに対してアルゴリズムが正の結果を返す raw カウント	センサーに触れたときの信号の 80%

パラメーター	説明	推奨値
ノイズしきい値	Figure 42 に示すように、ベースラインが更新されない raw カウント制限。 言い換えれば、raw カウントがベースライン + ノイズしきい値 を超えている限り、ベースラインは一定のままです。	信号の 30%。ほとんどの近接ソリューションでは、アルゴリズムを制御するために人間の手の動きの速度が遅くなるため、耐液性が必要な場合を除き、このしきい値をできるだけ低く保ちます。
負のノイズしきい値	raw カウント制限。この制限を下回ると、低ベースラインリセットパラメーターで指定されたサンプル数に対してベースラインが更新されなくなります。	信号の 40%
低ベースラインリセット	これらすべてのサンプルの raw カウントが異常に負のノイズしきい値を下回っている場合、ベースラインが現在の raw カウントにリセットされるサンプルの最大数。	デフォルト、つまり 30 のままにしておきます。
ヒステリシス	システムノイズによるセンサーステータス出力の切り替わりを防ぐために、以下で説明するしきい値に加えて使用される値。 センサーの状態が報告されます。 • 差分カウント > しきい値 + ヒステリシスの場合に ON。 • 差分カウント < しきい値 – ヒステリシスの場合は OFF。	信号の 10%
ON デバウンス	このパラメーターは、システムから ON 状態を生成するためにセンサーがアクティブでなければならない連続 CAPSENSE™スキャンの回数を示します。デバウンスにより、高周波、高振幅のノイズが誤検出を引き起こさないようにします。	3
マルチ周波数スキャン	マルチ周波数スキャンを有効にすると、CAPSENSE™コンポーネントは 3 つの異なるセンスクロック周波数でセンサー スキャンを実行し、対応する差分カウントを取得します。センサー差分カウントの中央値が、さらなる処理のために選択されます。	この機能は、特定のセンサースキャン周波数で外部ノイズが存在する場合でも堅牢な動作を実現するために使用します。 サンプルコード - CE237687 - PSoC™ 4: CAPSENSE™ multi frequency scan を参照してください。
CDAC 自動校正	この機能により、ファームウェアは、必要な校正目標の 70%を達成するために (初期化時に) CDAC を自動的に校正できます。	CDAC 自動校正は常に有効にすることを推奨します。

5 チューニングガイドライン

パラメーター	説明	推奨値
補償 CDAC を有効にする	補償コンデンサは、センサーからの過剰な寄生容量を補償して感度を高めるために使用されます。これを有効にすると信号が増加します。	C_P が低すぎる場合を除き、補償 CDAC を有効にすることを推奨します。
補償 CDAC 分周器	「センスクロック分周器」への分周器。単一センスクロックで補償コンデンサが切り替わる回数 (K_{comp}) を決定します。 $K_{comp} = \frac{Sense\ Clock\ Divider}{Comp\ CDAC\ Divider}$ 「CDAC 自動校正」が有効になっている場合、自動校正されます。	補償 CDAC 分周器の推奨値は 4 以上です。 K_{comp} が整数になるように値を設定します。
CDAC ディザを有効にする	入力容量がスイープされると、raw カウントは容量に応じて直線的に増加するはずです。raw カウントが入力容量に対して直線的に変化しない領域があり、これらはフラットスポットと呼ばれます。詳細については、フラットスポットのセクションを参照してください。ディザリングは、ディザ CDAC を使用してフラットスポットを減らすのに役立ちます。ディザ CDAC は、変換ポイントを平坦領域の周囲に移動させるホワイトノイズを追加します。	常に有効にすることを推奨します。
スキャン解像度 (第 4 世代 CAPSENSE™)	スキャン解像度は、CAPSENSE™のシグマデルタコンバータの解像度です。スキャン解像度はスキャン時間と感度を定義します。感度が向上すると、有効近接検知距離が長くなります。 第 4 世代 CAPSENSE™では 6 ビットから 16 ビットまで設定可能です。 第 5 世代 CAPSENSE™の場合、スキャン解像度は、変調器クロック, N_{SUB}, センスクロック, CIC2 フィルターの使用などの複数の要素に基づいて自動計算されます。	利用可能な最大数 (16)
raw カウント校正レベル (%)	–	70%

CAPSENSE™デザインのチューニングパラメーターは、ModusToolbox™の CAPSENSE™コンフィギュレータから調整できます。CAPSENSE™パラメーターの詳細については、AN85951 – PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™デザインガイドを参照してください。

5.3 耐液性のチューニング

耐液性に関するファームウェアのガイドラインで説明されているように、CAPSENSE™設計で耐液性を実装するには 2 つの推奨手法があります。ここでは、耐液性を向上させるための CAPSENSE™パラメーターのチューニングについて説明します。

5.3.1 CSD 技術による耐液性のチューニング

水滴やその他の液体が近接センサーに落ちると、誤ったトリガーが発生する可能性があります。液体の液滴が近接センサー (CSD を使用して実装) に落ちると、近接センサーの上に手をかざしたときに追加される静電容量と同等以上の静電容量が追加され、誤ったトリガーが発生する可能性があります。液滴による誤トリガーを排除するには、(必要な近接感知距離で) 近接センサーの上に手を置いたときの信号が少なくとも 3 倍大きくなるように CAPSENSE™ CSD パラメーターを調整することを推奨します。液滴の影響で信号よりも大きくなります。これにより、センサーはあらゆる状況で確実に動作することが保証されます。

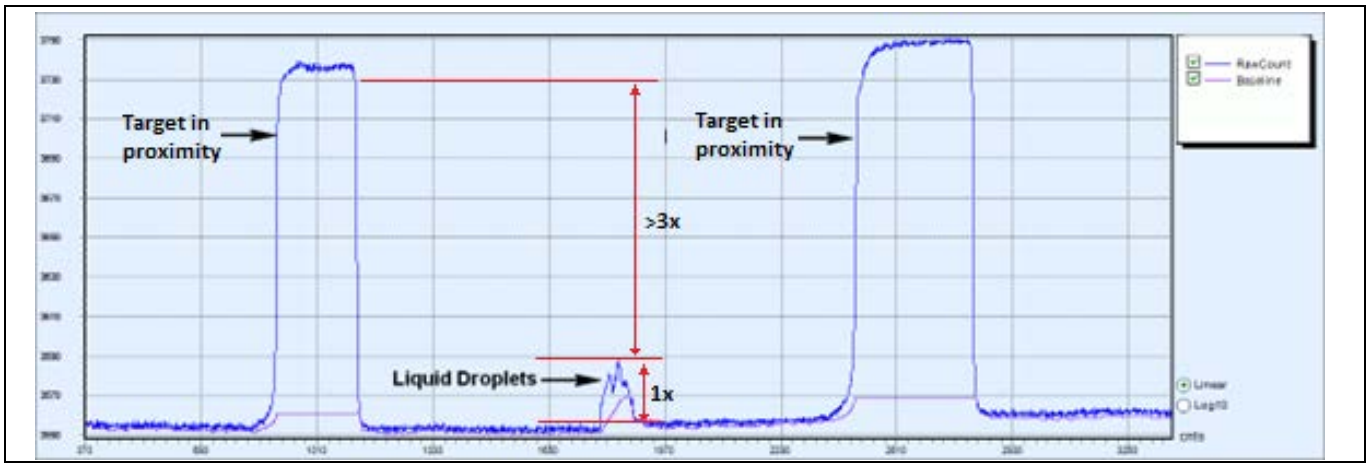


Figure 45 近接したターゲットオブジェクトによる信号と液滴による信号

Figure 46 に示すように、耐液性を考慮して近接センシング設計を調整するには、次の手順に従います。

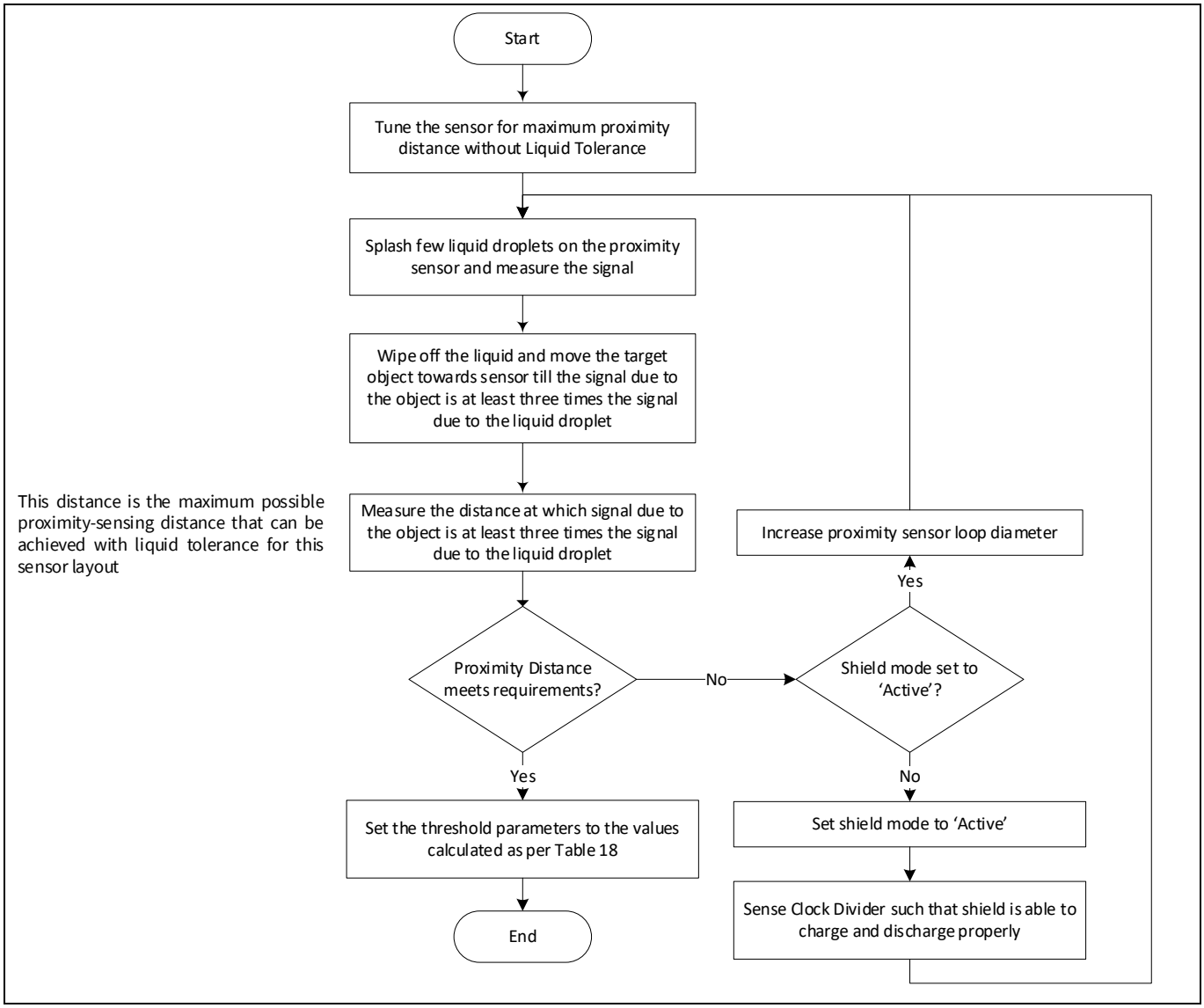


Figure 46　耐液性を考慮して近接センサーを調整するためのフローチャート

Table 18 に、CSD に基づく耐液性技術のチューニングパラメーターを示します。

Table 18　CSD の耐液性チューニングパラメーター

パラメーター	影響	トレードオフ	推奨値
シールドモード	アクティブシールドモードは、液体の存在により近くのグランドに結合するセンサーの磁力線の影響を軽減し、液体の存在により測定される raw カウントを減らします。	消費電力	“アクティブ”
センスクロック分周器	値を大きくすると、センスクロック周波数が低くなります。センスクロック周波数は、シールド容量が適切に充電および放電できるように十分に低い必要があります。	スキャン時間	シールド容量が適切に充電できない場合は、28 から始めて 4 の倍数に増やします。

5.3.2 CSX ガードベースの耐液性技術のチューニング

CSX ガードセンサー技術を液体の存在の検出に使用する場合は、Table 19 に示すようにパラメーターを設定します。

Table 19 CSX ガードの耐液性チューニングパラメーター

パラメーター	CSX センシングの説明と影響	推奨値
raw カウント校正レベル %	Raw カウント校正レベルは、現在のベースラインカウントが検出可能な最大静電容量のパーセンテージを示します。	50%
サブコンバージョン数 (N_{SUB})	N_{SUB} が高いほど分解能が高くなり、ガードセンサーの感度が高くなります。	近接センサー設定の値と同じにしておきます
指のしきい値	液滴の存在により raw カウントがベースラインよりも下に減少するため、正の raw カウント (ベースラインより上) には関係がなく、指のしきい値は無視できます。	–
ノイズしきい値	ベースラインを下回る raw カウント値のみが使用されるため、正のノイズしきい値は、システム内に存在するノイズに従ってベースラインが更新されることを保証するためのものです。	信号の raw カウントの 30%
負のノイズしきい値	ノイズしきい値が高くなると、存在する少量の液体に対する感度が低くなります。液体による信号がノイズしきい値を下回る場合、ベースラインが更新されるためです。	表面に液体が存在する場合、信号の raw カウントの 30%がベースラインより下で測定されます
低ベースラインリセット	このウィジェットはガードとして設定されているため、ベースラインよりも低い raw カウント値が液体の存在を検出するために使用されます。液体の存在により raw カウントが低下した場合に、ベースラインがまったくリセットされないようにこの値を設定します。	65535
ヒステリシス	ヒステリシス値は、raw カウントの変動によるトグルを回避するのに役立ちます	近接センサー設定の値と同じにしておきます
ON デバウンス	このパラメーターは、液体の存在を確認するために必要な連続した陽性結果の数を示します。高いほど良いですが、応答時間が長くなります	近接センサー設定の値と同じにしておきます
液体アクティブしきい値	これは、液体の存在を検出するために負の差分カウント (= raw カウント – ベースライン) と比較される値です。	この値は、センサー表面に液体が存在するときに測定される信号数に依存します。 この値はアプリケーションに実装する必要があります

5.4 内蔵セルフテスト (BIST)

CAPSENSE™ミドルウェア ライブラリは、CAPSENSE™近接センシングソリューションの設計, チューニング, およびデバッグに役立つ次の機能を可能にする BIST を提供します。

5.4.1 ハードウェアテスト

- センサーピンの完全性チェック
- 外部コンデンサとセンサーの静電容量測定
- VDDA 測定

5.4.2 ファームウェアテスト

- グローバルおよびウィジェット固有の設定の検証
- センサーのベースライン完全性チェック
- センサーの raw カウントとベースラインが指定された範囲内にある

BIST の実装および使用方法の詳細については、ModusToolbox™の Figure 47 に示す CAPSENSE™ミドルウェアライブラリのヘルプを参照してください。

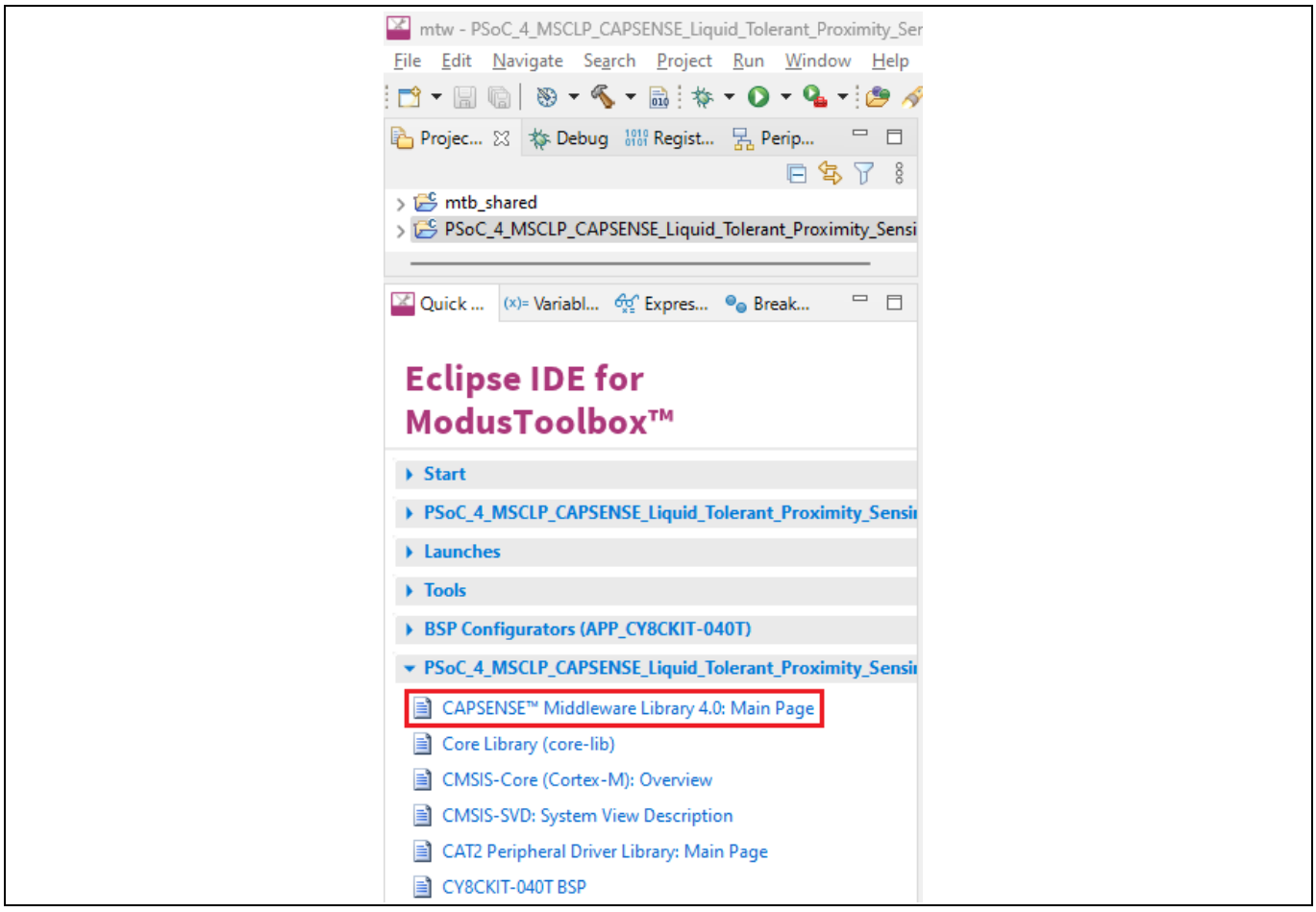


Figure 47 ModusToolbox™の CAPSENSE™ミドルウェアライブラリのヘルプ

5.5 トラブルシューティングのヒントとテクニック

トラブルシューティングとデバッグについては、AN85951 – PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™デザインガイドの「チューニング デバッグ FAQ」セクションを参照してください。

参考資料

アプリケーションノート:

[1] AN85951- PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™デザインガイド

[2] AN79953 - Getting started with PSoC™ 4

[3] AN234231 - Achieving lowest-power capacitive sensing with PSoC™ 4000T

[4] AN86233 - PSoC™ 4 MCU low-power modes and power reduction techniques

サンプルコード:

[5] PSoC™ 4: MSCLP CAPSENSE™ low-power proximity

[6] PSoC™ 4: CAPSENSE™ MSCLP liquid tolerant proximity sensing

[7] PSoC™ 4: MSCLP CAPSENSE™ low power

[8] PSoC™ 4: MSCLP robust low-power liquid-tolerant CAPSENSE™

その他のリソース:

[9] CAPSENSE™ webpage

[10] ModusToolbox™ webpage

[11] ModusToolbox™ software help on GitHub

改訂履歴

版数	発行日	変更内容
**	2021-10-22	本版は英語版 001-92239 Rev. *B 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。
*A	2024-01-18	これは英語版 001-92239 Rev. *D を翻訳した日本語版 002-34177 Rev. *A です。
*B	2024-08-14	これは英語版 001-92239 Rev. *F を翻訳した日本語版 002-34177 Rev. *B です。

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2024-08-14
Published by
Infineon Technologies AG
81726 Munich, Germany

© 2024 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?

Email: erratum@infineon.com

Document reference
002-34177 Rev. *B

重要事項

本手引書に記載された本製品の使用に関する手引きとして提供されるものであり、いかなる場合も、本製品における特定の機能性能や品質について保証するものではありません。本製品の使用の前に、当該手引書の受領者は実際の使用環境の下であらゆる本製品の機能及びその他本手引書に記された一切の技術的情報について確認する義務が有ります。インフィニオンテクノロジーズはここに当該手引書内で記される情報につき、第三者の知的所有権の不侵害の保証を含むがこれに限らず、あらゆる種類の一切の保証および責任を否定いたします。

本文書に含まれるデータは、技術的訓練を受けた従業員のみを対象としています。本製品の対象用途への適合性、およびこれら用途に関連して本文書に記載された製品情報の完全性についての評価は、お客様の技術部門の責任にて実施してください。

警告事項

技術的要件に伴い、製品には危険物質が含まれる可能性があります。当該種別の詳細については、インフィニオンの最寄りの営業所までお問い合わせください。
インフィニオンの正式代表者が署名した書面を通じ、インフィニオンによる明示の承認が存在する場合を除き、インフィニオンの製品は、当該製品の障害またはその使用に関する一切の結果が、合理的に人的傷害を招く恐れのある一切の用途に使用することはできないこと予めご了承ください。