

PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™ デザインガイド

About this document

Scope and purpose

CAPSENSE™ デザインガイドは、PSoC™ 4 および PSoC™ 6 MCU デバイスファミリでの CAPSENSE™ 機能を使って静電容量タッチセンシングアプリケーションを設計する方法について説明します。CAPSENSE™ 機能は、前例のない信号対雑音比 (SNR)、クラス最高の耐液性、およびボタン、スライダー、タッチパッド、および近接センサーなどのさまざまなセンサーを提供します。このデザインガイドでは、CAPSENSE™ の操作、CAPSENSE™ 設計ツール、PSoC™ Creator および ModusToolbox™ CAPSENSE™ コンポーネントのパフォーマンス調整、および設計上の考慮事項について説明します。このガイドでは、前世代のデバイスに比べていくつかの利点がある第 5 世代の CAPSENSE™ テクノロジーについても紹介します。

さまざまなデバイスファミリが CAPSENSE™ 機能で利用できます。特定のデバイスを選択していない場合、または静電容量センシングを初めて使用する場合は、[Getting started with CAPSENSE™ design guide](#) を参照してください。メカニカルボタンに対する CAPSENSE™ の利点、CAPSENSE™ テクノロジーの基礎を理解し、アプリケーションに適したデバイスを選択するのに役立ちます。また、設計に役立つ適切なドキュメント、キット、またはツールを紹介します。

Intended audience

本ドキュメントは、主に PSoC™ 4 および PSoC™ 6 MCU デバイスの CAPSENSE™ 設計原理に精通する必要があるエンジニアを対象とします。

Table of contents

Table of contents

1	はじめに	7
1.1	概要	7
1.2	CAPSENSE™ の特長	7
1.3	PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™ Plus の特長	8
1.4	CAPSENSE™ の設計フロー	9
2	CAPSENSE™ 技術	12
2.1	CAPSENSE™ の基礎	12
2.1.1	自己容量センシング	13
2.1.2	相互容量センシング	15
2.2	静電容量タッチセンシング方式	16
2.2.1	CAPSENSE™ シグマデルタ (CSD)	16
2.2.2	CAPSENSE™ クロスポイント (CSX)	17
2.3	信号対ノイズ比 (SNR)	18
2.4	CAPSENSE™ のウィジェット	19
2.4.1	ボタン (0 次元)	19
2.4.2	スライダー (1 次元)	22
2.4.3	タッチパッド/トラックパッド (2 次元)	23
2.4.4	近接 (3 次元)	23
2.5	耐液性	24
2.5.1	自己容量センシングの耐液性	25
2.5.1.1	自己容量センサーに対する液滴と液体流の影響	25
2.5.1.2	被駆動シールド信号およびシールド電極	27
2.5.1.3	ガードセンサー	28
2.5.2	相互容量センシングの耐液性	29
2.5.2.1	相互容量センサーに対する液滴と液体の流れの影響	29
2.5.2.2	相互容量センサーの耐液性のための自己容量センシングの使用	30
2.5.3	耐液性性能への液体属性の影響	31
3	PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™	33
3.1	PSoC™ 4 および PSoC™ 6 の CAPSENSE™ 世代	33
3.2	CAPSENSE™ CSD センシング方式	35
3.2.1	GPIO セルの静電容量-電流変換器	36
3.2.2	IDAC ソースモード	37
3.2.3	IDAC シンクモード	38
3.2.4	CAPSENSE™ クロックジェネレータ	39
3.2.4.1	センスクロック	39
3.2.4.2	変調クロック	40
3.2.5	シグマデルタ変換器	40
3.2.6	アナログマルチプレクサ (AMUX)	42
3.2.7	CAPSENSE™ CSD シールド	42
3.3	CAPSENSE™ CSX センシング方式 (第 3 および第 4 世代)	43
3.4	CAPSENSE™ CSD-RM センシング方式 (第 5 世代)	45
3.4.1	GPIO セルの容量から電荷へのコンバーター	46
3.4.2	コンデンサ DAC (CDAC)	46
3.4.3	CAPSENSE™ クロックジェネレーター	47
3.4.3.1	センスクロック	47
3.4.3.2	変調クロック	47
3.4.4	レシオメトリックセンシングテクノロジー	47
3.4.5	アナログマルチプレクサ (AMUX) と制御マトリックス (CTRLMUX)	48

Table of contents

3.4.6	CAPSENSE™ CSD-RM シールド	49
3.4.6.1	アクティブシールド	49
3.4.6.2	パッシブシールド	49
3.5	CAPSENSE™ CSX-RM センシング方式 (第 5 世代).....	50
3.5.1	レシオメトリックセンシングテクノロジー	51
3.6	自律スキャン.....	52
3.7	マルチチャネルの使用.....	52
4	CAPSENSE™設計および開発ツール	54
4.1	PSoC™ Creator	54
4.1.1	CAPSENSE™コンポーネント	54
4.1.2	CapSense_ADC コンポーネント	55
4.1.3	チューナーGUI	55
4.1.4	サンプルプロジェクト.....	55
4.2	ModusToolbox™	56
4.2.1	CAPSENSE™ ミドルウェア	56
4.2.2	CAPSENSE™設定ツール	56
4.2.3	CSDADC ミドルウェア.....	57
4.2.4	CSDIDAC ミドルウェア.....	57
4.2.5	CAPSENSE™チューナー	57
4.2.6	サンプルプロジェクト.....	57
4.3	ハードウェアキット.....	58
5	CAPSENSE™性能のチューニング	60
5.1	SmartSense および手動チューニングの選択.....	60
5.2	SmartSense	60
5.2.1	概要	60
5.2.2	SmartSense の全自動チューニング	62
5.2.2.1	チューニングボタンウィジェット	62
5.2.2.2	チューニング スライダー ウィジェット	65
5.2.2.3	チューニング近接ウィジェット	66
5.2.3	SmartSense ハードウェア パラメーター - Only モード.....	66
5.2.4	初期チューニングの SmartSense	66
5.3	手動チューニング	67
5.3.1	概要	67
5.3.2	CSD センシング方式 (第 3 および第 4 世代).....	68
5.3.2.1	基本	68
5.3.2.2	CAPSENSE™ハードウェアパラメーターの選択	74
5.3.2.3	CAPSENSE™ソフトウェアパラメーターの選択	80
5.3.2.4	ボタンウィジェット調整.....	86
5.3.2.5	スライダー ウィジェット調整.....	87
5.3.2.6	タッチパッドウィジェット調整	90
5.3.2.7	近接ウィジェット調整	92
5.3.3	CSX センシング方式 (第 3 および第 4 世代).....	93
5.3.3.1	基本	93
5.3.3.2	CAPSENSE™ハードウェアパラメーターの選択	94
5.3.3.3	CAPSENSE™ソフトウェアパラメーターの選択	96
5.3.3.4	ボタンウィジェット調整.....	96
5.3.3.5	タッチパッドウィジェットの調整	98
5.3.4	CSD-RM センシング方式 (第 5 世代).....	101
5.3.4.1	基本	101
5.3.4.2	CAPSENSE™ハードウェアパラメーターの選択	107
5.3.4.3	CAPSENSE™ソフトウェアパラメーターの選択	116

Table of contents

5.3.4.4	自律スキヤンの設定	117
5.3.4.5	マルチチャネルスキヤニング	120
5.3.4.6	ボタンウィジェット調整	121
5.3.4.7	スライダーウィジェット調整	122
5.3.4.8	タッチパッドウィジェット調整	122
5.3.4.9	近接ウィジェットの例	123
5.3.5	CSX-RM センシング方式 (第 5 世代).....	123
5.3.5.1	基本	123
5.3.5.2	CAPSENSE™ハードウェアパラメーターの選択	125
5.3.5.3	CAPSENSE™ソフトウェアパラメーターの選択	130
5.3.5.4	自律スキヤンの構成	130
5.3.5.5	マルチチャネルスキヤン	130
5.3.5.6	ボタンウィジェットの調整	130
5.3.5.7	タッチパッドウィジェットのチューニング	131
5.3.6	手動チューニングのトレードオフ.....	132
5.3.6.1	信頼性	133
5.3.6.2	消費電力および応答時間	133
5.3.7	チューニング デバッグ FAQ.....	133
5.3.7.1	チューナーがデバイスと通信しない.....	133
5.3.7.2	チューナーを介してデバイスのパラメーターを更新できない.....	134
5.3.7.3	デバイスに接続できますが、raw カウントが表示されない	134
5.3.7.4	差分カウントはセンサーに指を置いた時に、10 から 20 カウントまで少しだけ変化する	134
5.3.7.5	システムのチューニングした後、テスト中に大量の放射ノイズがある	135
5.3.7.6	手動チューニング後、スキヤン時間はシステム要件を満たさない.....	135
5.3.7.7	システムは 85%に校正できない	135
5.3.7.8	スライダーの重心応答が非線形である.....	136
5.3.7.9	スライダー セグメントは大きな変動のある C_p がある.....	136
5.3.7.10	GPIO を切り替えたとき raw カウントがレベルシフトまたはノイズの増加を示す	136
5.3.7.11	SNR が低くなっている	138
5.3.7.12	CSX ボタンの C_M が低いことを確認	139
6	CAPSENSE™のジェスチャー.....	142
6.1	タッチジェスチャーの対応.....	142
6.2	ジェスチャーグループ.....	142
6.3	1 本指ジェスチャーの実装.....	142
6.3.1	ウィジェットの調整.....	142
6.3.2	事前定義されたジェスチャーの選択.....	143
6.3.3	タイムスタンプを使用したファームウェアの実装.....	144
6.3.4	ジェスチャーパラメーターの調整.....	144
6.3.4.1	ジェスチャー パラメーターを調整するためのチューナーGUI の使用.....	144
6.3.4.2	クリック	146
6.3.4.3	スクロール	148
6.3.4.4	1 本指のフリック	150
6.4	2 本指のジェスチャーの実装.....	150
6.5	ジェスチャーの高度なフィルター.....	151
7	設計上の注意事項.....	152
7.1	ファームウェア.....	152
7.1.1	低消費電力の設計.....	153
7.2	センサーの構造.....	155
7.3	オーバーレイの選択.....	156
7.3.1	オーバーレイの素材.....	156

Table of contents

7.3.2	オーバーレイの厚さ.....	156
7.3.3	オーバーレイの接着剤.....	157
7.4	PCB レイアウト ガイドライン	157
7.4.1	センサーC _p	157
7.4.2	回路基板の層	158
7.4.3	ボタン デザイン.....	158
7.4.3.1	自己容量のボタン デザイン.....	158
7.4.3.2	相互静電容量ボタンの設計	159
7.4.4	スライダの設計.....	166
7.4.4.1	スライダ セグメントの形状、幅およびエアギャップ.....	167
7.4.4.2	スライダの端におけるダミー セグメント.....	171
7.4.4.3	スライダ寸法の決定	172
7.4.4.4	スライダ セグメント配線の経路.....	172
7.4.4.5	LED 付きスライダ デザイン	173
7.4.5	センサーとデバイスの配置.....	173
7.4.6	配線の長さ と幅	174
7.4.7	配線の経路	174
7.4.8	クロストークに対するソリューション.....	175
7.4.9	ビア	176
7.4.10	グランド面	176
7.4.10.1	E-pad のないパッケージの使用.....	178
7.4.10.2	E-pad のあるパッケージの使用.....	179
7.4.10.3	PSoC™ 4 Bluetooth® LE デバイスの使用	179
7.4.11	電源レイアウトの推奨事項.....	180
7.4.12	耐液性のレイアウト ガイドライン.....	181
7.4.12.1	シールド電極のレイアウトガイドライン.....	182
7.4.12.2	ガード センサーのレイアウトガイドライン.....	183
7.4.12.3	グランドリングによる耐液性.....	184
7.4.13	回路図のルール チェックリスト.....	185
7.4.13.1	外部コンデンサ接続のピン選択	185
7.4.13.2	センサーピンの選択	186
7.4.14	レイアウト ルール チェックリスト	188
7.5	CAPSENSE™ システムのノイズ	191
7.5.1	Finger Injected ノイズ.....	191
7.5.1.1	Finger injected ノイズを減らすための推奨事項.....	192
7.5.2	VDDA ノイズ	193
7.5.2.1	VDDA ノイズを低減するための推奨事項	193
7.5.3	外部ノイズ	193
7.5.3.1	ESD 保護.....	193
7.5.3.2	電磁適合性 (EMC) に対する考慮事項.....	195
7.6	接地の影響.....	206
7.6.1	CSX 方式.....	206
7.6.1.1	CbodyDG>>Cfs.....	206
7.6.1.2	CbodyDG<<Cfs.....	207
7.6.2	CSD 方式.....	207
7.6.2.1	AC/DC 駆動のアプリケーション	208
7.6.2.2	バッテリー駆動のアプリケーション	208
8	CAPSENSE™ Plus	210
9	リソース	214
9.1	ウェブサイト.....	214
9.2	デバイス データシート	214

Table of contents

9.3	コンポーネント データシート/ミドルウェア文書.....	214
9.4	テクニカル リファレンス マニュアル	214
9.5	開発キット.....	214
9.6	PSoC™ Creator.....	214
9.7	ModusToolbox™	214
9.8	アプリケーション ノート	215
9.9	設計サポート	215
10	用語集	216
	改訂履歴	222

はじめに

1 はじめに

1.1 概要

静電容量タッチセンサーは、人体の静電容量を使ってセンサー上またはセンサー近くの指の存在を検出するユーザーインターフェースのデバイスです。CAPSENSE™ソリューションは、ユーザーの製品に優雅で、信頼性が高く、使いやすい静電容量タッチセンシング機能をもたらします。

本デザインガイドでは、PSoC™ 4 および PSoC™ 6 MCU デバイスファミリの CAPSENSE™機能を中心に説明します。これらのデバイスは、単一のチップに構成可能なアナログとデジタル周辺機能、メモリ、無線およびマイクロコントローラーを集積している真のプログラマブル組み込みシステムオンチップです。また、これらのデバイスは、非常に柔軟性が高く、CAPSENSE™の他に ADC、DAC および Bluetooth® LE などの多くの機能を備えており、市場投入までの時間を短縮、重要なシステム機能を統合、そしてシステム全体のコストを削減します。

本ガイドは、読者がサイプレスの PSoC™ Creator 統合開発環境 (IDE) を使用して PSoC™ 4 と PSoC™ 6 MCU 向けのアプリケーション開発に慣れていることを想定しています。PSoC™ 4 が初めての方は [AN79953 - Getting started with PSoC™ 4](#) または [AN92167 - Getting started with PSoC™ 4 Bluetooth® LE](#) を参照してください。PSoC™ 6 MCU が初めての方は [AN221774 - Getting started with PSoC™ 6 MCU](#) および [AN210781 - Getting started with PSoC™ 6 MCU with Bluetooth® LE connectivity](#) 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。

ModusToolbox™が初めての方は [ModusToolbox™ IDE quick start guide](#) を参照してください。

本デザインガイドは、以下の項目について説明します。

- [PSoC™ 4 および PSoC™ 6 MCU の CAPSENSE™テクノロジー](#)
- [PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™で利用可能な設計および開発ツール](#)
- [PSoC™ 4 および PSoC™ 6 MCU の CAPSENSE™ PCB レイアウトガイドライン](#)
- [PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™コンポーネントのパフォーマンスチューニング](#)
- [モーター制御システムや IH 調り器などの CAPSENSE™ Plus 機能を使用するアプリケーション](#)

1.2 CAPSENSE™の特長

PSoC™ 4 および PSoC™ 6 MCU の CAPSENSE™は以下の特長があります。

- すべての CAPSENSE™対応 GPIO ピン¹で自己容量 (CSD) および相互容量 (CSX) ベースのタッチセンシングのサポート
- クラス最高の[信号対ノイズ比](#)を提供し、高感度を可能にし、広範囲の近接センシング (最大 30cm の近接センシング距離) と耐液性動作を提供 ([耐液性](#)を参照)
- さまざまなオーバーレイ素材と厚さに対応する高性能センシング ([CAPSENSE™の基礎, オーバーレイの素材](#), および[オーバーレイの厚さ](#)を参照してください)
- [SmartSense](#) 自動チューニング技術
- 電磁妨害 (EMI) をさらに低減するための疑似ランダム シーケンス (PRS) のクロックソース, スペクトラム拡散およびプログラム可能な抵抗スイッチをサポート
- ハイバネートモード時の低消費電力 (最小 1.71V の動作および最小 150nA の消費電流)

PSoC™ 4100S Max デバイスは、第 5 世代の CAPSENSE™テクノロジー ([レシオメトリックセンシングテクノロジー](#)) を導入し、旧世代と比較して次の追加機能を備えます。

¹ 最高の CAPSENSE™パフォーマンスを実現するためには、[センサーピンの選択](#)の推奨事項に従ってください。

はじめに

- **SNR の改善:** 第 5 世代 CAPSENSE™ テクノロジー (**レシオメトリックセンシングテクノロジー**) は、前世代のデバイスと比較してノイズパフォーマンスを大幅に改善します。
- **リフレッシュレートの向上:** マルチセンスコンバーター (MSC) の感度が向上すると、前世代と同様の信号を取得するために必要な時間が短縮されるため、より高いリフレッシュレートを実現できます。センサーを並行してスキャンできる 2 つの独立した MSC ブロックは、特に多数のセンサーをスキャンするユースケースで、リフレッシュレートをさらに向上させます。
- **CPU 帯域幅の改善:** スキャンは CPU モードと DMA モードの両方でサポートされています。CPU モードは従来の割込み駆動モードですが、DMA モードは自律スキャンが可能であり、CPU 帯域幅の要件を前世代と比較して 18% 削減します。
- **ノイズ耐性の向上:** レールツーレールスイングがセンス電圧として使用されます。これにより、最大のセンス電圧が提供され、耐性が向上します。第 5 世代 CAPSENSE™ テクノロジーでは、全波差動センシングが自己容量センシングに使用され、これにより、外部環境からセンサルーティングに誘導されるノイズが相殺されます。このセンシング技術は、電源 (V_{DD}) ノイズに対する耐性も優れています。

1.3 PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™ Plus の特長

静電容量タッチセンシングと追加のシステム機能を備えた PSoC™ 4 **CAPSENSE™ Plus アプリケーション** が作成できます。CAPSENSE™ の他に、これらデバイスの主な特長は以下のとおりです。

- シングルサイクルの乗算器を実装した Arm® Cortex®-M0/M0+ CPU を備え、48MHz で最大 43DMIPS を実現
- -40°C~85°C の周囲温度範囲における 1.71V~5.5V の動作
- 最大 128KB のフラッシュ (CM0+は 8 ビット分解能で 2 倍以上のコード密度を持つ)
- 最大 16KB の SRAM
- 最大 94 個のプログラマブル GPIO
- 補完的なデッドバンド プログラマブル出力を備えた独立した中央揃え PWM、同期 ADC 動作 (PWM サイクル内でユーザー指定可能時間に ADC をトリガーする能力)、および同期リフレッシュ (異常な波形を回避するために、全 PWM にわたり PWM のデューティ比の変更を同期化する能力)
- PWM キル信号 (過剰電流が検出された場合、モーター駆動を停止) のコンパレータベースのトリガー
- ゼロオーバーヘッドシーケンシング付きのサンプルホールド (S&H) 機能を備えた 12 ビットの 1Msps ADC。これにより、全 ADC 帯域幅を信号変換に使用し、シーケンサーのオーバーヘッドに使用しないことが可能
- コンパレータモードと SAR 入力バッファリング機能付きのオペアンプ
- 最大 4 つの COM 電極をサポートするセグメント LCD ダイレクトドライブ
- SPI/UART/I2C シリアル通信チャネル
- バージョン 4.0 に準拠した Bluetooth® LE 通信およびバージョン 4.1 の複数の機能
- ユニバーサル デジタル ブロック (UDB) と呼ばれるプログラマブルロジックブロック。各々は 8 個のマクロセルとカスケード接続可能なデータパスを持ち、プログラマブルペリフェラル (I2S など) の効率的な実装が可能
- コントローラー エリア ネットワーク (CAN)
- 完全にサポートされた PSoC™ Creator の設計入力、開発、デバッグ環境は以下を提供:
 - 設計入力とビルド (アナログ配線を含む)
 - すべての固定機能のペリフェラルおよび一般的なプログラマブルペリフェラル用のコンポーネント
 - ドキュメントおよびトレーニング モジュール
- MDK Arm® 環境 (以前、RealView と知られている) と別の環境へのビルドの移植に対応
- ModusToolbox™ 用の Eclipse 統合開発環境 (IDE) に対応

はじめに

CAPSENSE™の他に、PSoC™ 6 MCU デバイスの主な特長は以下のとおりです。

- 単一の CPU デバイス (Arm® Cortex®-M4)、デュアル CPU デバイス (Arm® Cortex®-M4 および Cortex®-M0+) ハードウェアでプロセッサ間通信をサポート
- 1.1V または 0.9V でユーザー選択可能なコア ロジック動作ができる 1.71V~3.6V のデバイス動作電圧
- 最大 2MB のフラッシュメモリと最大 1MB の SRAM
- アナログ、デジタル、CAPSENSE™またはセグメント LCD 機能のために使用される最大 78 個の GPIO
- プログラマブル アナログ ブロック: 2 つのオペアンプ、構成可能な PGA、コンパレータ、12 ビット 1Msps SAR ADC、12 ビット電圧モード DAC
- プログラマブル デジタル ブロック、通信インターフェース
- 16 ビット/32 ビット タイマー、カウンター、PWM または直交デコーダーに構成可能な 12 の UDB、32 の TCPWM
- I2C、SPI または UART インターフェースに構成可能な最大 13 個のシリアル通信ブロック (SCB)。詳細については[デバイスデータシート](#)を参照してください。
- 1 つの I2S インターフェースと 2 つの PDM チャンネルを備えたオーディオサブシステム
- 外部クアッド SPI フラッシュメモリからの直接実行 (execute-in-place) とオンザフライの暗号化および復号をサポートする SMIF インターフェース
- Bluetooth® LE 5.0 規格準拠の Bluetooth® Smart 接続 (Bluetooth® LE 接続搭載の PSoC™ 6 MCU デバイスファミリーにのみ適用可能)

要件に合わせて適切な CAPSENSE™デバイスを選択するために、[AN64846 - Getting started with CAPSENSE™](#)を参照してください。

1.4 CAPSENSE™の設計フロー

Figure 1 に静電容量センシングを使用する製品設計サイクルの一般的なフローを示します。本ガイドに記載されている情報は緑色でハイライト表示されています。**Table 1** に、**Figure 1** で番号付けた各タスクをサポートするドキュメントへのリンクを示します。

はじめに

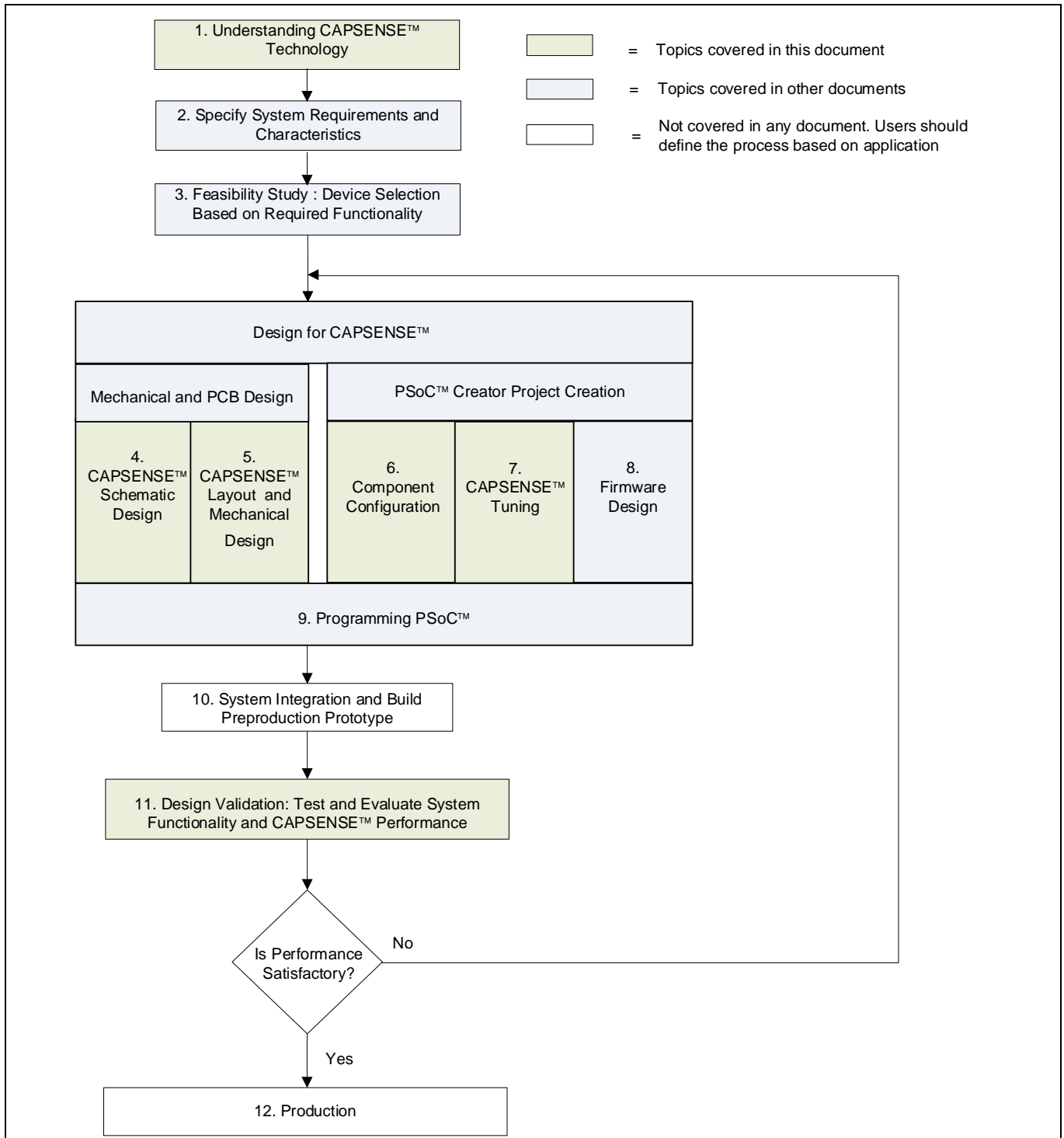


Figure 1 CAPSENSE™製品の設計フロー

はじめに

Table 1 サポートしているドキュメント

フローチャート内の ステップ	対応する資料	
	ドキュメント名	章
1. CAPSENSE™の理解	CAPSENSE™ デザイン ガイド (本書) Getting started with CAPSENSE™	第 2 章および第 3 章 -
	Getting started with CAPSENSE™	-
2. 要件定義	Getting started with CAPSENSE™	-
3. 実行可能性調査	PSoC™ 4 データシート PSoC™ 4 Bluetooth® LE データシート PSoC™ 6 MCU データシート	-
	AN64846 - Getting started with CAPSENSE™ design guide	-
	AN79953 - Getting started with PSoC™ 4	-
	AN91267 - Getting started with PSoC™ 4 Bluetooth® LE AN221774 - Getting started with PSoC™ 6 MCU	-
4. 回路図設計	CAPSENSE™ デザイン ガイド (本書)	第 7 章
5. レイアウト設計	CAPSENSE™ デザイン ガイド (本書)	第 7 章
6. コンポーネントの 構成	PSoC™ CAPSENSE™ コンポーネント データシート/ミドルウェア文書	-
	CAPSENSE™ デザイン ガイド (本書)	第 5 章
7. 性能チューニング	CAPSENSE™ デザイン ガイド (本書)	第 5 章
8. ファームウェア設 計	PSoC™ コンポーネント データシート/ミドルウェア文書	-
	PSoC™ Creator サンプルプロジェクト	-
	こちらから ModusToolbox™ をダウンロードできます。 ModusToolbox™ の関連文書を参照してください。 ModusToolbox™ release notes ModusToolbox™ user guide ModusToolbox™ quick start guide ModusToolbox™ CAPSENSE™ configurator guide ModusToolbox™ CAPSENSE™ tuner guide PSoC™ Creator to ModusToolbox™ porting guide	-
9. PSoC™ プログラミ ング	IDE プログラミング用の PSoC™ Creator ユーザー ガイド スタンドアロン プログラミング用の PSoC™ Programmer ホーム ページ および MiniProg3 user guide	-
10. 試作	-	-
11. 設計検証	CAPSENSE™ デザイン ガイド (本書)	第 5 章
12. 量産	-	-

CAPSENSE™ 技術

2 CAPSENSE™ 技術

静電容量タッチセンシング技術は、センサー上またはセンサー近くの指の存在を検出するために、面(センサー)とその周辺環境との間で起きる静電容量の変化を測定します。

2.1 CAPSENSE™ の基礎

標準的な CAPSENSE™ センサーは、PCB 表面にエッチングされた適切な形状とサイズの銅パッドから成ります。Figure 2 に示すように、非導電性のオーバーレイは、ボタンのタッチ面として機能します。

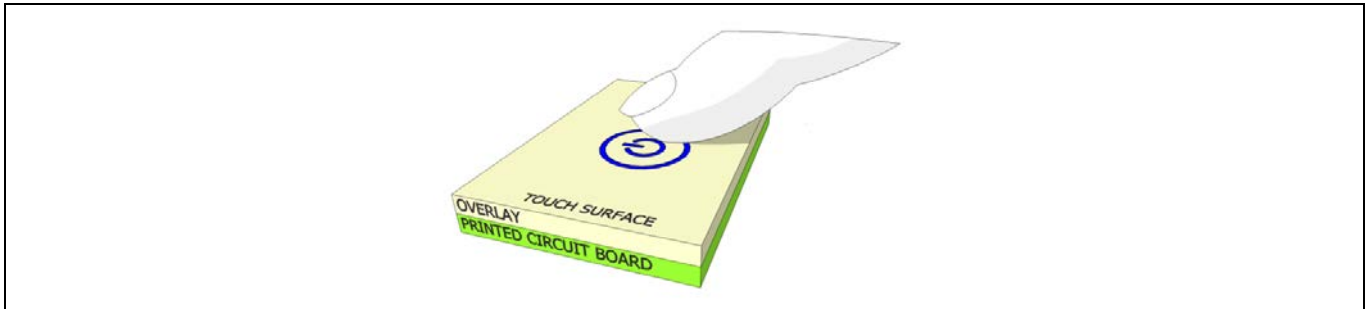


Figure 2 静電容量タッチセンサー

PCB 配線とビアは、センサーパッドを CAPSENSE™ のセンサーピンとして構成されている PSoC™ GPIO に接続します。Figure 3 に示すように、各電極の自己容量は C_{SX} として、電極間の相互容量は C_{MX} としてモデル化されます。PSoC™ 内の CAPSENSE™ 回路で、これら静電容量の値を等価デジタルカウントに変換します(詳細は第 3 章を参照してください)。これらのデジタルカウントはタッチを検出するために CPU で処理されます。

また CAPSENSE™ は、自己容量センシング用および相互容量センシング用に外部コンデンサ C_{MOD} または C_{INT} を必要とします。第 3 および第 4 世代 CAPSENSE™ アーキテクチャでは、自己容量センシングには単一の C_{MOD} コンデンサが必要であり、相互容量センシング用には C_{INTA} および C_{INTB} コンデンサが必要です。シールド電極は、耐液性または遠く離れた近接センシングのために実装される場合、追加の C_{TANK} コンデンサが必要とされる場合があります。第 5 世代の CAPSENSE™ アーキテクチャの場合、各チャンネルの自己容量と相互容量の両方のセンシングに 2 つの C_{MOD} 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 ピンとグラウンドの間に接続されます。Table 34 に、外部コンデンサの推奨容量値を示します。

CAPSENSE™ 技術

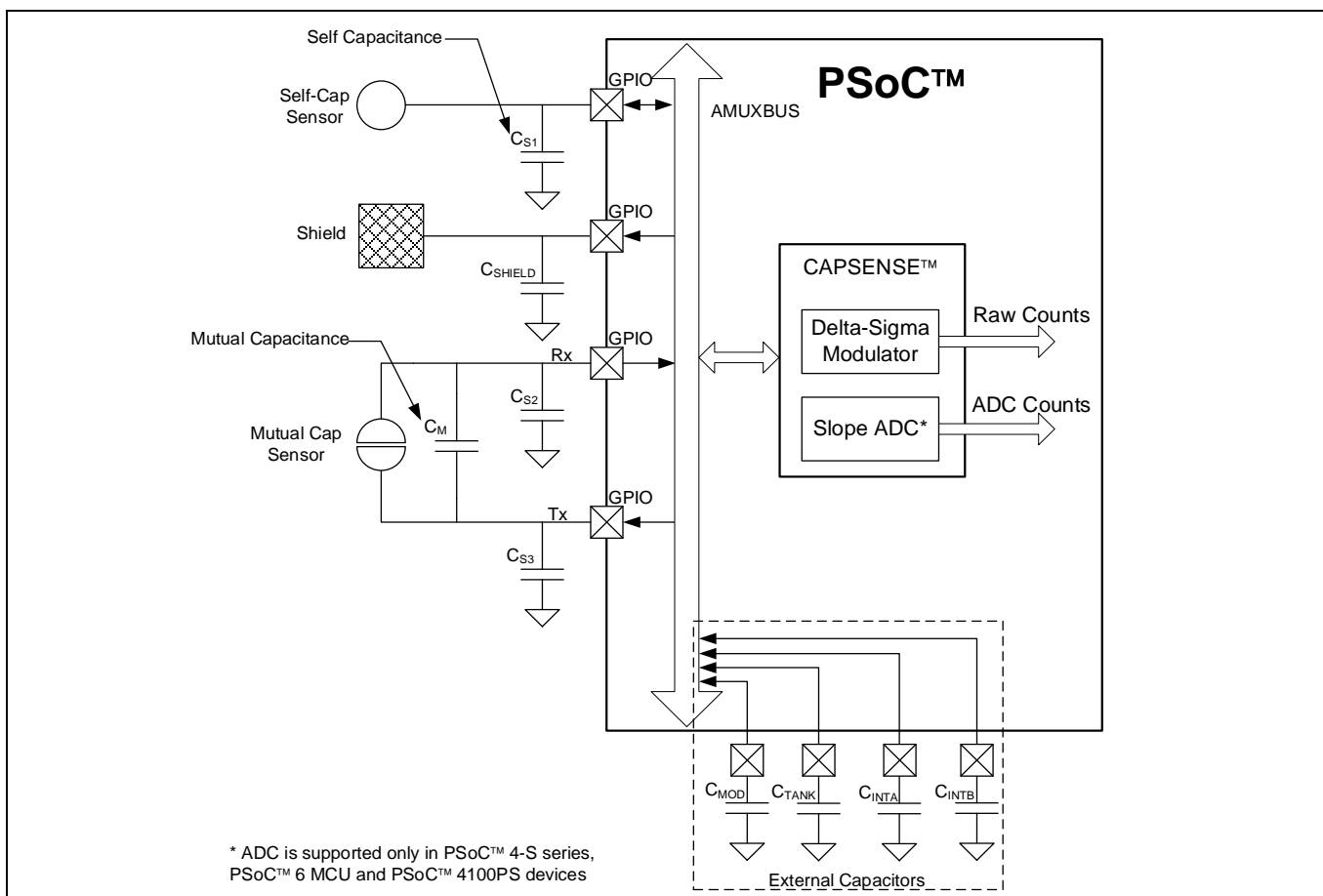


Figure 3 PSoC™ デバイス, センサー, および外部コンデンサ

指を置かない場合のセンサーの静電容量は寄生容量 C_p と呼ばれます。 C_p は、センサー (センサーパッド、配線、ビアを含む) とシステム内の他の導電体 (グランド面、配線、および製品のシャーシやエンクロージャ内の金属など) との間で生じる電場により生じます。 PSoC™ の GPIO と内部容量も寄生容量の一因となります。しかし、これらの内部容量はセンサー容量に比べて非常に小さいです。

2.1.1 自己容量センシング

Figure 4 は、自己容量センシングのために GPIO ピンを配線とビアでセンサーパッドに接続する方法を示します。一般的には、グランド (GND) ハッチがセンサーパッドを他のセンサーと配線から分離させるためにセンサーパッドの周りを囲んでいます。 Figure 4 にセンサーパッド周辺でいくつかの磁力線を示しますが、実際の電界分布は非常に複雑です。

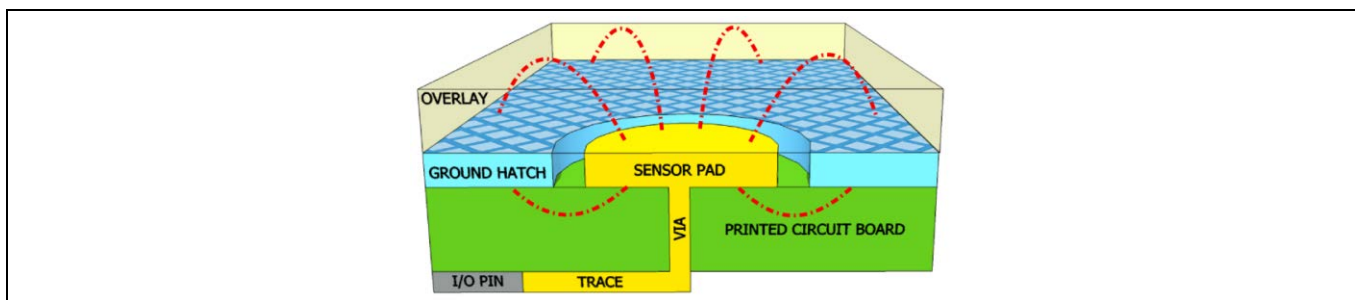


Figure 4 寄生容量

CAPSENSE™ 技術

Figure 5 に示すように、指がオーバーレイに触れると、人体の導電性と大きな体積はセンサーパッドに平行な接地された導電性の面と成ります。

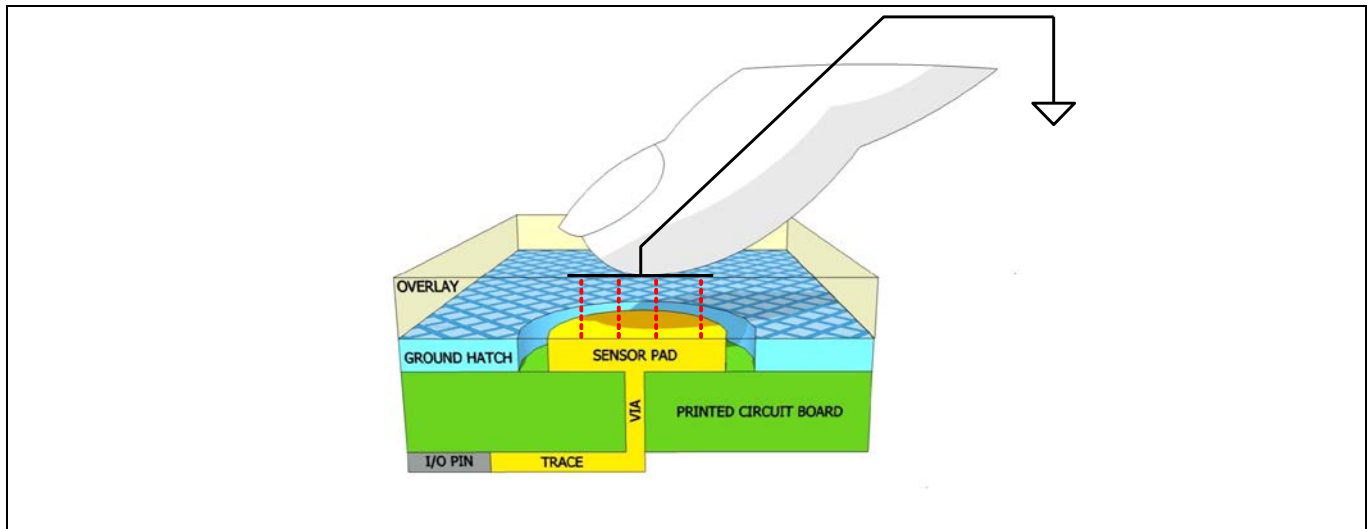


Figure 5 指の静電容量

ここでの構造は平行板のコンデンサと成ります。センサーパッドと指の間の静電容量を式 1 に示します。

式 1. 指の静電容量

$$C_F = \frac{\epsilon_0 \epsilon_r A}{d}$$

ここで、

ϵ_0 = 真空の誘電率

ϵ_r = オーバーレイの比誘電率

A = 指とセンサーパッドが重なっている面積

d = オーバーレイの厚さ

C_F = 指の静電容量

C_p と C_F は互いに並列であり、これは、両者がセンサーピンとグランド間の容量を表すためです。そのため、指がセンサーに触れているときのセンサーの総静電容量 C_S は、 C_p と C_F の和になります。

式 2. 指がセンサーに触れているときのセンサーの総静電容量

$$C_S = C_p + C_F$$

タッチしていない場合は、 C_S は C_p です。

PSoC™ は、容量 C_S を raw カウントと呼ばれる等価デジタルカウントに変換します。指のタッチにより、センサーピンの総静電容量が増加するため、raw カウントの増加は指のタッチを示します。推奨される SNR を達成でき、特定のデバイスでサポートされている C_p 範囲については、[デバイスデータシート/コンポーネントデータシート/ミドルウェア文書](#) の CSD 仕様を参照してください。

CAPSENSE™ 技術

2.1.2 相互容量センシング

Figure 6 に相互容量センシングのボタンセンサーレイアウトを示します。相互容量センシング方式は、送信 (Tx) 電極と受信 (Rx) 電極と呼ばれる 2 本の電極間の容量を測定します。

相互容量センシングシステムでは、VDDIO² または VDDD³ (VDDIO がデバイスでサポートされない場合) と GND の間に切り替えるデジタル電圧信号が Tx ピンに印加され、Rx ピンで受信した電荷量が測定されます。Rx 電極で受信された電荷量は 2 本の電極間の相互容量 (C_M) に比例します。

Figure 7 に示すように、Tx と Rx 電極の間に指を置くと、相互容量は C_M^1 に低下します。相互容量の低下に起因して、Rx 電極で受信した電荷量も低下します。CAPSENSE™ システムは Rx 電極で受信した電荷量を測定して、タッチ有り/タッチ無し状態を判定します。

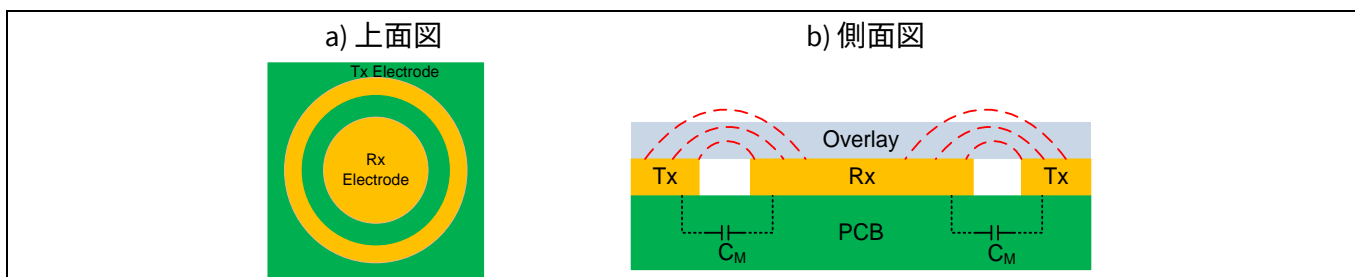


Figure 6 相互容量センシング動作

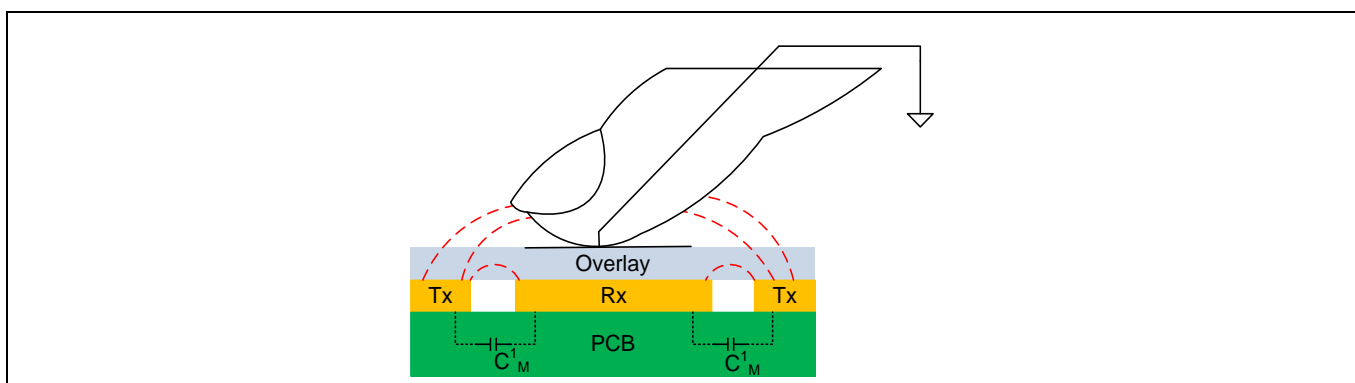


Figure 7 指のタッチがあるときの相互容量

²VDDIO は I/O ピンに電源を供給します。

³VDDD はデジタルセクションに電源を供給します。

CAPSENSE™ 技術

2.2 静電容量タッチセンシング方式

PSoC™ は、自己容量センシング用に CAPSENSE™ シグマデルタ (CSD)、相互容量スキャン用に CAPSENSE™ クロスポイント (CSX) として知られる特許取得の静電容量タッチセンシング方式を使用しています。CSD および CSX タッチセンシング方式は、クラス最高の **信号対ノイズ比 (SNR)** を提供します。これらのセンシング方式はハードウェアとファームウェアの技術を結合したものです。

2.2.1 CAPSENSE™ シグマデルタ (CSD)

Figure 8 に CSD 方式を簡略化したブロック図を示します。

CSD では各 GPIO が C_s を等価な電流に変換するスイッチトキャパシタ回路を持ちます。そして、アナログ MUX (AMUX) は、複数のセンサー電流の 1 つを選択して、それを電流-デジタル変換器へ供給します。電流-デジタル変換器は、デルタシグマ ADC に似ています。電流-デジタル変換器の出力カウント (**raw カウント**として知られる) は、電極間の自己容量に比例したデジタル値です。

式 3. CSD 内の raw カウントとセンサー静電容量の関係

$$\text{raw カウント} = G_{\text{CSD}} C_s$$

ここで、

G_{CSD} = CSD の静電容量-デジタル変換利得

C_s = 電極の自己容量

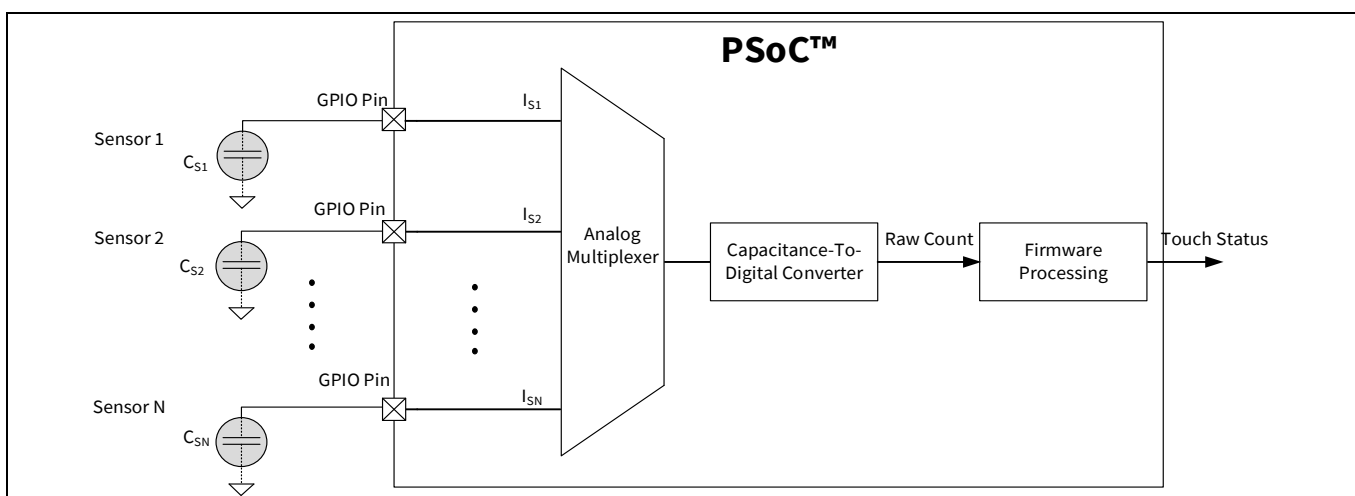


Figure 8 CSD 方式の概略図

Figure 10 に時間経過に対する raw カウントの変化を示します。指がセンサーに触れるとき、 C_s は C_p から $(C_p + C_F)$ に増加し、raw カウントも増加します。所定のしきい値と raw カウントの変化を比較することで、ファームウェアのロジックはセンサーがアクティブ (指がセンサーに触れる) であるかどうかを判定します。

CAPSENSE™ 技術

2.2.2 CAPSENSE™ クロスポイント (CSX)

Figure 9 は CSX 方式を簡略化したブロック図を示します。

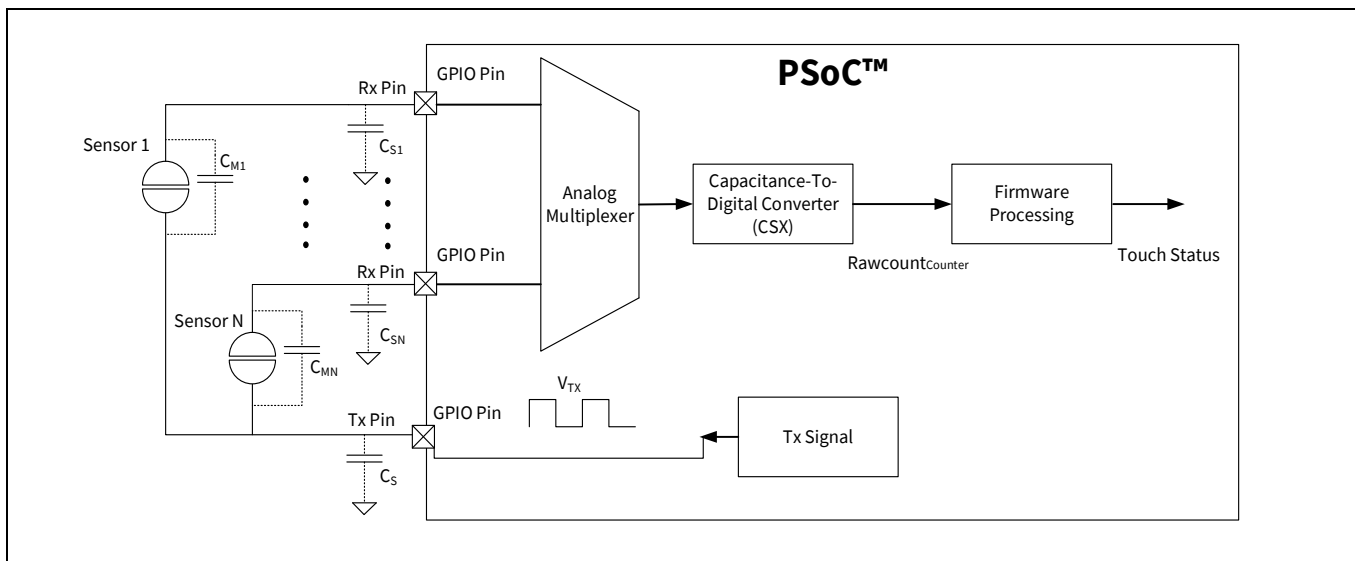


Figure 9 CAPSENSE™ クロスポイント (CSX) 方式の概略図

CSX を使用することで、Tx 電極の電圧は Rx 電極に電荷量をカップリングします。この電荷量は、Tx と Rx の電極間の相互容量に比例します。そして、アナログ MUX は、複数の Rx 電極 1 つを選択し、それを電流-デジタル変換器へ供給します。

Rawcount_{Counter} として知られる電流からデジタルへの変換器の出力カウントは、式 4 に示すように Rx 電極と Tx 電極間の相互静電容量に比例したデジタル値です。

式 4. CSX の raw カウントとセンサー静電容量の関係

$$\text{Rawcount}_{\text{Counter}} = G_{\text{CSX}} C_M$$

ここで、

G_{CSX} = 相互容量方式の静電容量-デジタル変換利得

C_M = 2 本の電極間の相互容量

Figure 10 に時間経過に対する raw カウントの変化を示します。指がセンサーに触れるとき、 C_M は C_M^1 から C_M^2 に低下するため (Figure 7 を参照してください)、カウンター出力も低下します。ファームウェアは、 C_M が低下すると raw カウントが増加するように正規化します。これは、CSD と CSX の方式で raw カウントを同じ視覚表現として維持します。所定のしきい値と raw カウントの変化を比較することで、ファームウェアのロジックはセンサーがアクティブ (指がセンサーに触れる) であるかどうかを判定します。正規化された逆 raw カウントは、式 15 を使用して計算されます。

CAPSENSE™ 技術

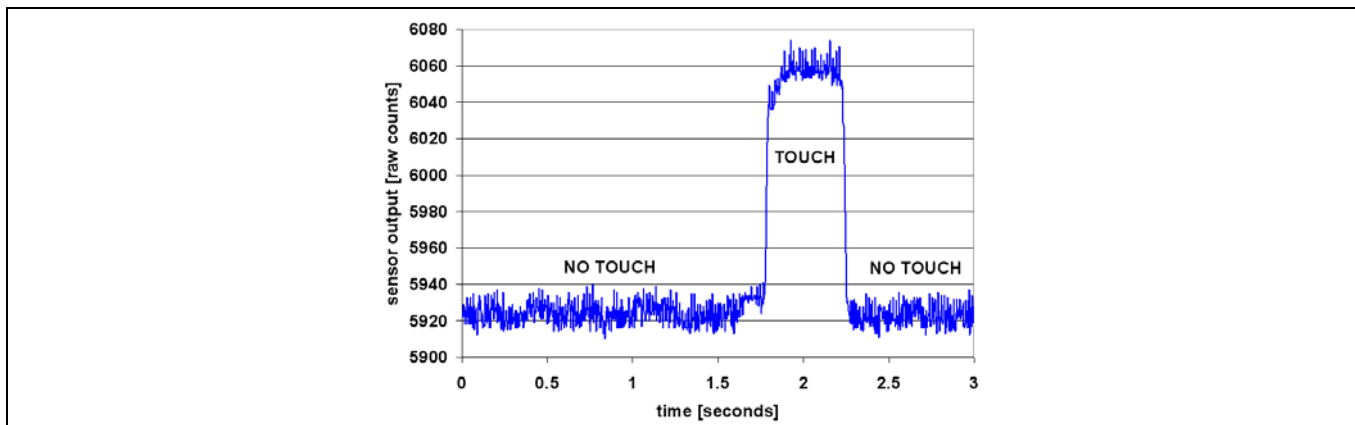


Figure 10 raw カウントと時間

PSoC™ 4 および PSoC™ 6 CAPSENSE™ CSD および CSX ブロックの詳細は [PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™](#) を参照してください。

2.3 信号対ノイズ比 (SNR)

実際には、raw カウントはシステム内の固有ノイズによって異なります。Figure 11 に示すように、CAPSENSE™ ノイズは非接触状態での raw カウントのピークツーピーク変動量です。

適切に調整された CAPSENSE™ システムでは、センサーのオンとオフ状態が確実に識別されます。良い性能を得るためには、CAPSENSE™ 信号は CAPSENSE™ ノイズより十分に大きくする必要があります。SNR は、CAPSENSE™ 信号の CAPSENSE™ ノイズに対する比と定義されており、CAPSENSE™ センサーの最も重要な性能パラメータです。

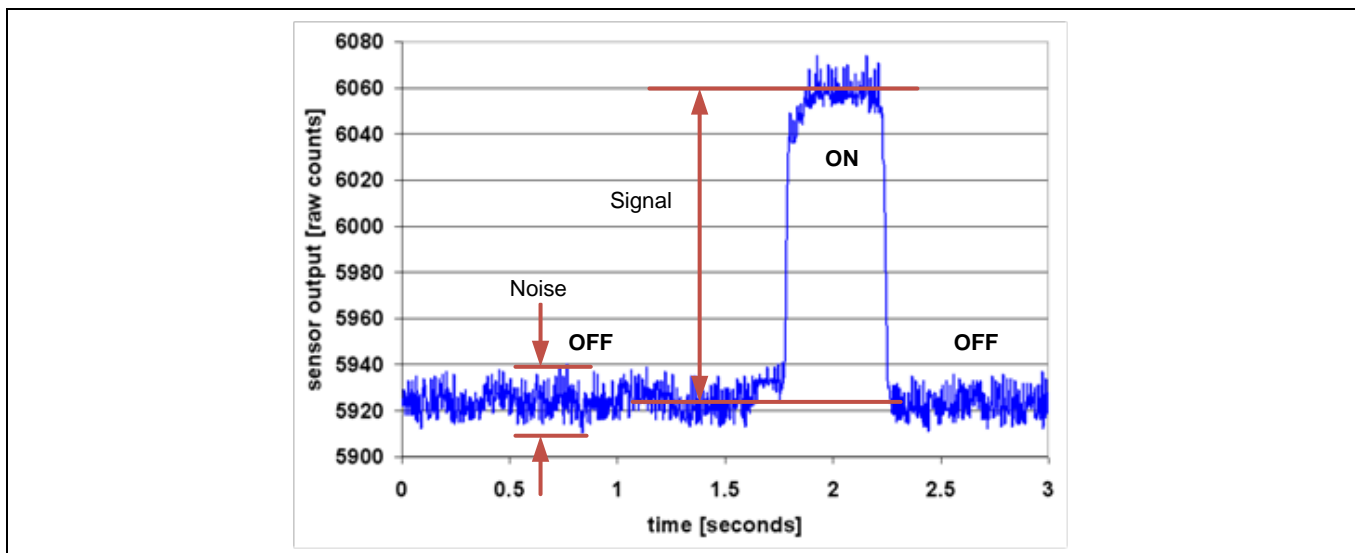


Figure 11 SNR

この例では、指が触れない場合の raw カウントの平均レベルは 5925 カウントです。指がセンサーに触れたとき、平均 raw カウントは 6060 カウントに増加するため、信号は $6060 - 5925 = 135$ カウントです。オフ状態では raw カウントの最小値は 5912 で、最大値は 5938 です。そのため、CAPSENSE™ ノイズは $5938 - 5912 = 26$ カウントです。結果として、 $135 / 26 = 5.2$ の SNR になります。

CAPSENSE™ 技術

CAPSENSE™ センサーの SNR の推奨最小値は 5 です。この 5:1 の比率は信頼性の高いオン/オフ動作を提供するために信号とノイズの間に十分なマージンを可能にするベスト プラクティスのしきい値設定からのものです。

2.4 CAPSENSE™ のウィジェット

CAPSENSE™ のウィジェットは 1 つ以上の CAPSENSE™ センサーから構成しており、ユーザー インターフェースの特定の種類を表します。CAPSENSE™ のウィジェットは大まかに 4 つのカテゴリに分類されます。ボタン (0 次元)、スライダー (1 次元)、タッチパッド/トラックパッド (2 次元)、および近接センサー (3 次元) です。Figure 12 にボタン、スライダーおよび近接センサーのウィジェットを示します。ここでは、異なる CAPSENSE™ ウィジェットの基本的な概念について説明します。センサー構造の詳細は [センサーの構造](#) を参照してください。

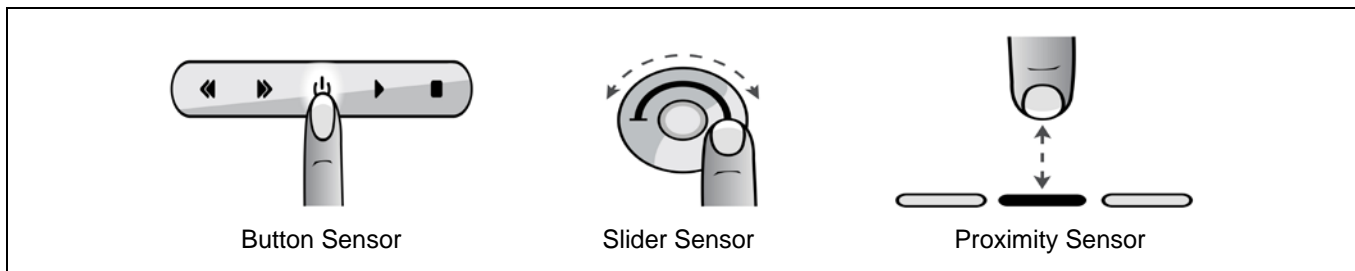


Figure 12 ウィジェットの種類

2.4.1 ボタン (0 次元)

CAPSENSE™ のボタンは、家電、医療機器、白物家電、照明管理や他の多くの製品の用途に幅広く使われているメカニカルボタンに取って代わります。これは、単一センサーから成る、CAPSENSE™ ウィジェットの最も簡単な種類です。CAPSENSE™ のボタンは、2 つの出力状態の内の 1 つを出力します。アクティブ (指がボタンに触れた) か、非アクティブ (指がボタンに触れない) です。これら 2 状態はそれぞれオンとオフ状態とも呼ばれます。

自己容量センシング方式 (CSD) の場合、簡単な CAPSENSE™ ボタンは PCB 配線で PSoC™ GPIO に接続された円形の銅パッドで構成されます。ボタンは他のボタンや配線から分離するために接地した銅のハッチで囲まれています。円形のギャップは、ボタンパッドとグランドハッチを分離しています。各ボタンは 1 つの PSoC™ GPIO を必要とします。これらのボタンは、非導電性基板上的の任意の導電性材料を使用して構築できます。例えば、ガラス基板上的のインジウムスズ酸化物、または非導電性フィルム上の銀インクです。金属製のバネでもボタンセンサーとして使用できます。詳細については、[センサーの構造](#) を参照してください。

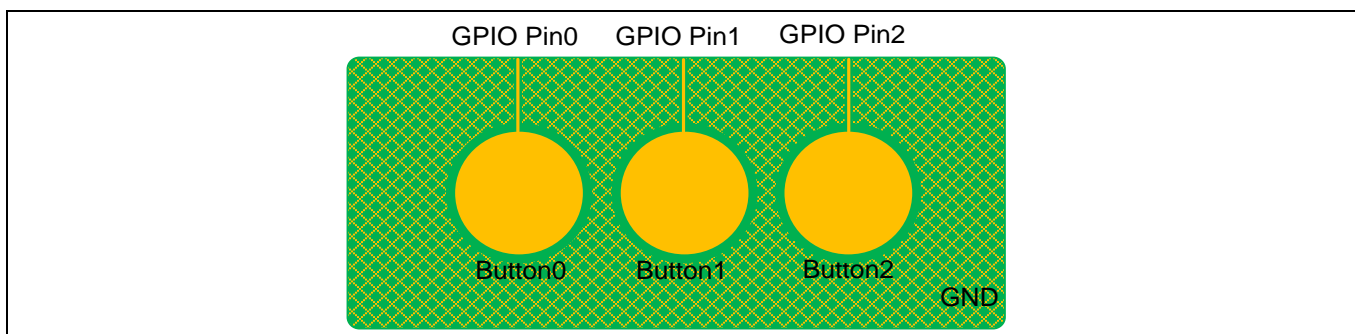


Figure 13 簡単な CAPSENSE™ ボタン

CAPSENSE™ 技術

相互容量センシング方式 (CSX) の場合、各ボタンは Tx 電極として構成された GPIO ピンと Rx 電極として構成された GPIO ピンを必要とします。Figure 14 に示すように、Tx は複数のボタン間で共有できます。

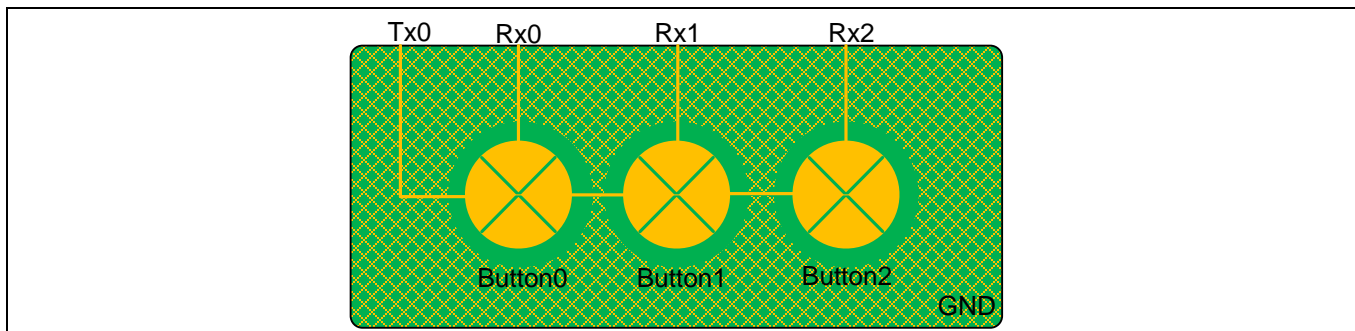


Figure 14 相互容量センシング方式での簡単な CAPSENSE™ ボタン

Figure 15 に示すように、多くのボタンを必要とする応用製品 (例えば電卓のキーパッドや QWERTY キーボードなど) の場合、CAPSENSE™ ボタンをマトリクス状に配置できます。これにより、GPIO ごとに複数のボタンを持つ設計を可能にします。例えば、Figure 15 に示す 16 ボタンの設計は 8 本の GPIO のみを必要とします。

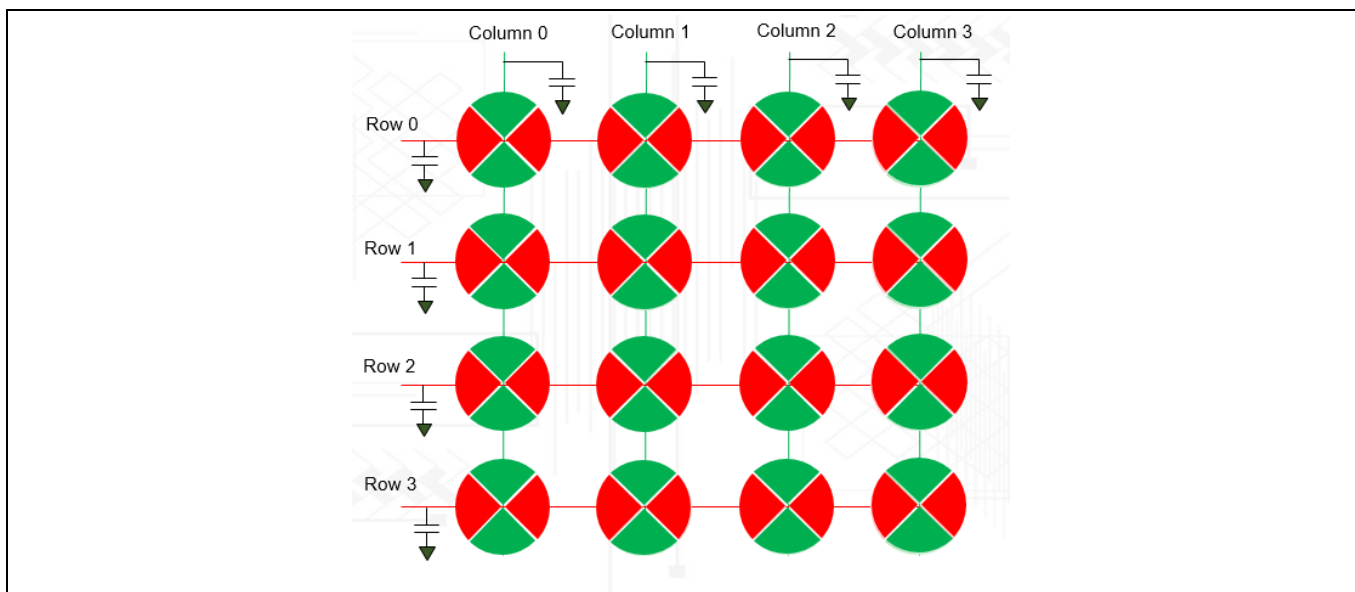


Figure 15 CSD を基にしたマトリクス ボタン

マトリクス ボタンの設計は、行センサーと列センサーという 2 つの静電容量センサーのグループがあります。マトリクス ボタンのアーキテクチャは、自己容量 (CSD) および相互容量 (CSX) の両方の方式で使用できます。

Figure 15 に示すように、CSD モードでは各ボタンが行センサーと列センサーで構成されます。ボタンに触れると、行センサーと列センサーの両方がアクティブになります。CSD ベースのマトリクスボタンは、ユーザーが 1 度に 1 つのボタンに触れることが予想される場合にのみ使用する必要があります。ユーザーが対角線上にある複数のボタンに触れた場合、Figure 16 に示すように、指の位置を解決できません。この効果はゴースト効果と呼ばれ、無効な状態と見なされます。

CAPSENSE™ 技術

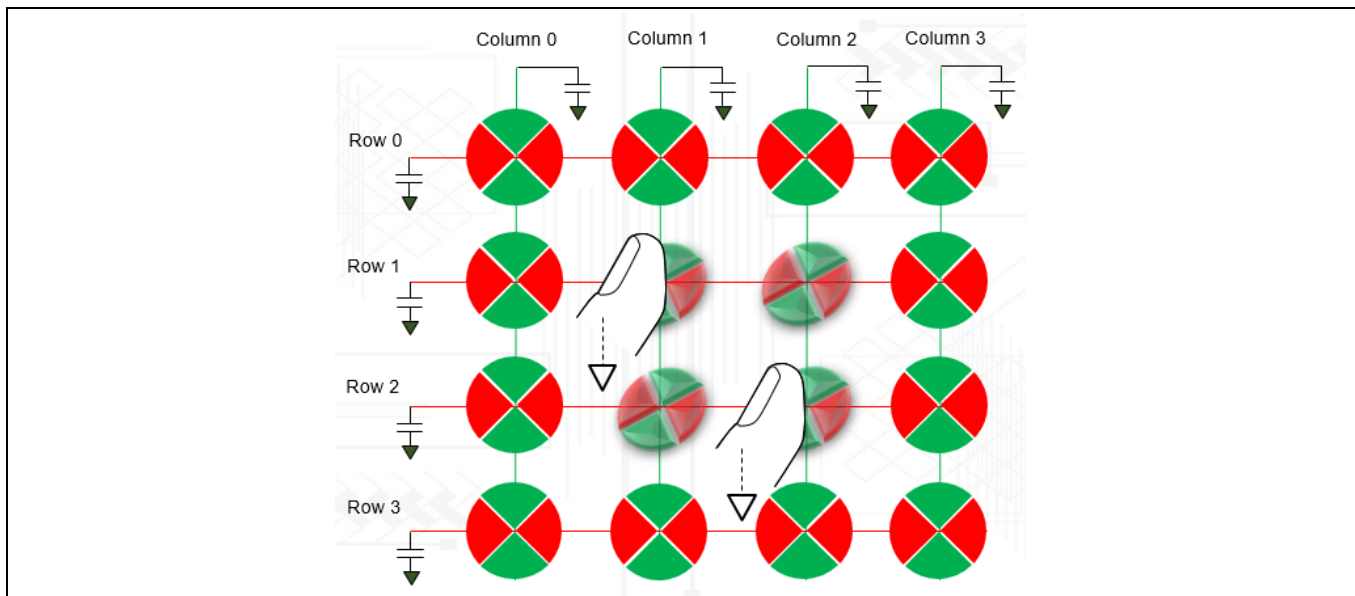


Figure 16 CSD を基にしたマトリックス ボタンのゴースト効果

相互静電容量は、ゴーストタッチ現象の影響を受けず、高 Cp センサーの SNR が向上する方法であるため、マトリックスボタンの推奨されるセンシング方法です。これは、Figure 17 に示すように、行と列を検出するのではなく、各交差点で形成される相互容量をセンシングします。Shift, Ctrl, および Alt キーを備えたキーボードなど、複数のボタンを同時にセンシングする必要があるアプリケーションでは、CSX センシング方式を使用できます。または、Shift, Ctrl, および Alt キーを個別の CSD ボタンとして設計する必要があります。

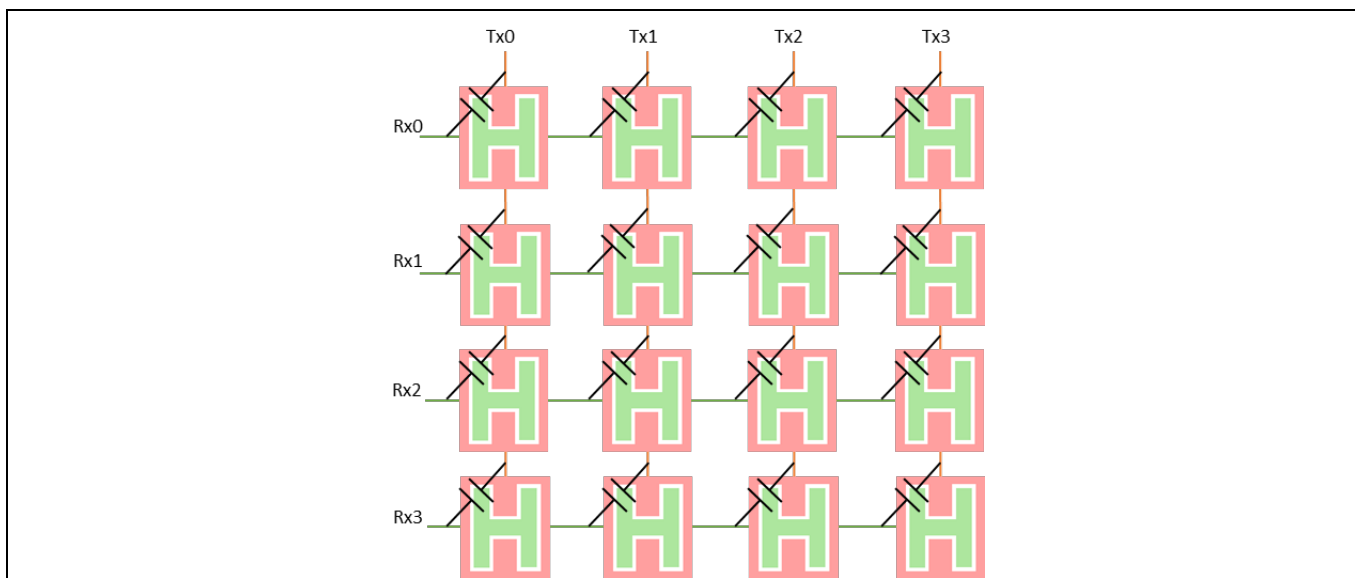


Figure 17 CSX を基にしたマトリックス ボタン

Note: CSX センシング方式を使用してマトリックスキーパッドをスキャンすると、CSD センシング方式よりも全体的なスキャン時間が長くなる場合があります。これは、CSD センシング方式が行と列をセンサーとしてスキャンするために対し、CSX センシング方式は各交差点をセンサーとしてスキャンするためです。

CAPSENSE™ 技術

2.4.2 スライダー (1 次元)

スライダーは、必要な入力が漸次インクリメントまたはデクリメントする形態である場合に使用されます。例として、照明管理 (調光装置), 音量管理, グラフィックイコライザ, 速度制御などが挙げられます。現在、PSoC™ Creator の CAPSENSE™ コンポーネントおよび ModusToolbox™ は自己容量ベーススライダーのみに対応します。相互容量ベーススライダーはコンポーネントの今後のバージョンで対応します。

スライダーは、互いに隣接する静電容量センサーの配列で構成されます。これらの配列はセグメントと呼ばれます。1 個のセグメントに触れると、隣接セグメントも部分的に活性化します。ファームウェアは、**重心位置**として知られる指のタッチの幾何学的中心の位置を計算するために、触れたセグメントと近隣のセグメントからの raw カウントを処理します。

計算された重心位置の実際の分解能は、スライダーのセグメント数よりも高いです。例えば、5 個のセグメントを持つスライダーでは、少なくとも 100 ケ所の物理的な指の位置を判別できます。この高精度の分解能は、スライダーを横切る指の滑りとして重心位置のスムーズな移動を感知します。

Figure 18 に示すように、リニア スライダーにおいては、セグメントがインラインに配置されます。各スライダーセグメントは PSoC™ GPIO に接続します。スライダーのセグメント用には、ジグザグのパターン (二重の山形) が推奨されます。このレイアウトにより、セグメントに触れると、必ず隣接セグメントにも部分的に触れます。これは、重心位置を推定する助けになります。

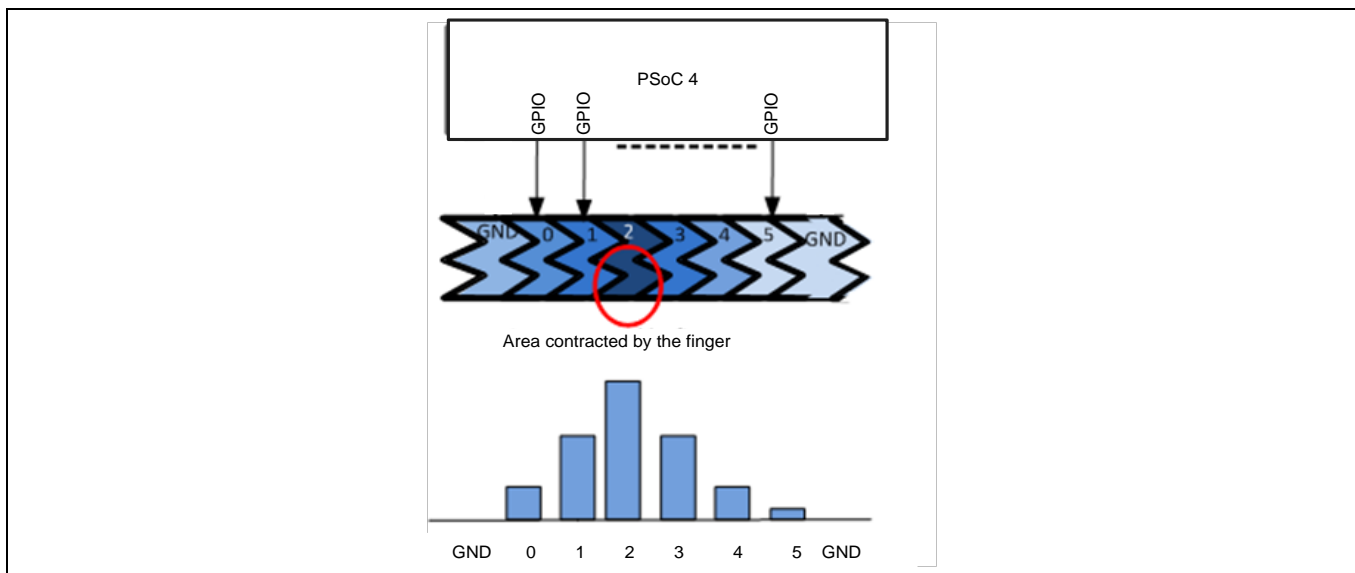


Figure 18 リニア スライダー

ラジアル スライダーは連続していることを除き、リニア スライダーに似ています。**Figure 19** に、一般的なラジアル スライダーを示します。

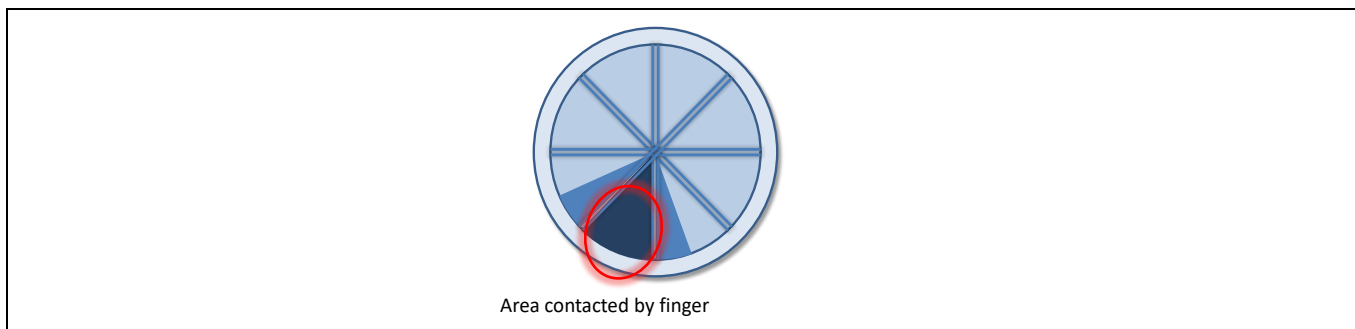


Figure 19 ラジアル スライダー

CAPSENSE™ 技術

2.4.3 タッチパッド/トラックパッド (2次元)

タッチパッドは、(トラックパッドとしても知られる)は、Xの長さおよびYの長さの両方で指の位置を特定できるように、XとYのパターンで配置された2個のリニアスライダがあります。Figure 20に、タッチパッドセンサーの典型的な配置を示します。マトリックスボタンと同様に、タッチパッドもCSDまたはCSXのいずれかの検知方法を使用して検知できます。

CSDベースのタッチパッドはゴーストタッチの影響を受けるため、シングルポイントタッチアプリケーションのみをサポートします。

現在、PSoC™ CreatorのCAPSENSE™コンポーネントは自己容量ベースのタッチパッドのみをサポートします。相互容量ベースのタッチパッドはコンポーネントの今後のバージョンでサポートします。

CSXタッチパッドはマルチポイントタッチアプリケーションをサポートできますが、この方法では行と列ではなく各交差点をスキャンするため、CSDタッチパッドと比較してスキャン時間が長くなる可能性があります。

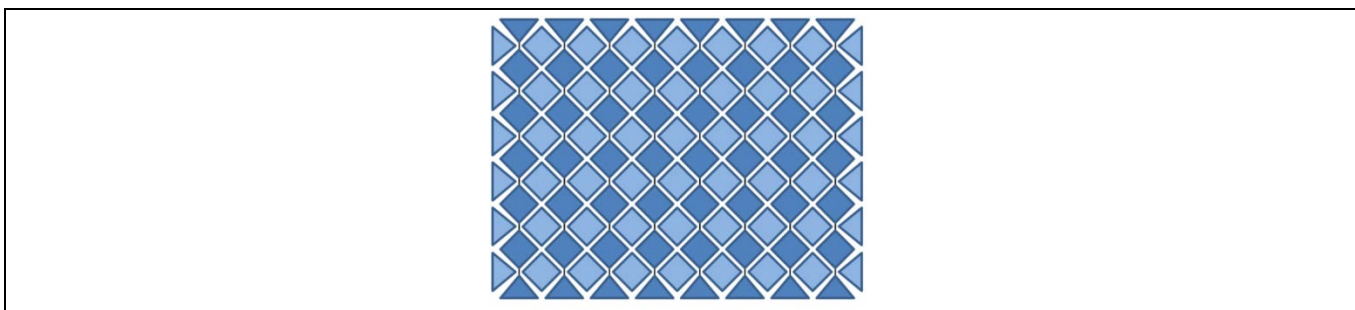


Figure 20 タッチパッドセンサーの配置

2.4.4 近接 (3次元)

近接センサーは、センサー周囲の3次元空間における手の存在を検出します。しかし、近接センサーの実際出力は、CAPSENSE™のボタン同様にオン/オフ状態です。近接センシングは、センサー構造により、数cm～数十cm離れて手を検出できます。自己容量は、近接アプリケーションで推奨されるセンシング方法です。

近接センシングには、ボタンやスライダの使用時に比べてはるかに長い距離に放射される電界が必要です。これは大きなセンサー面積を必要とします。しかし、大きなセンサー面積は、大きな寄生容量 C_p になり、検出がより困難になります。これは、小面積ながらも、長い距離で高電界強度を持つセンサーが必要です。Figure 21に、他のセンサーを囲む2～3mmの厚さのトレースを使用した近接センサーを示します。

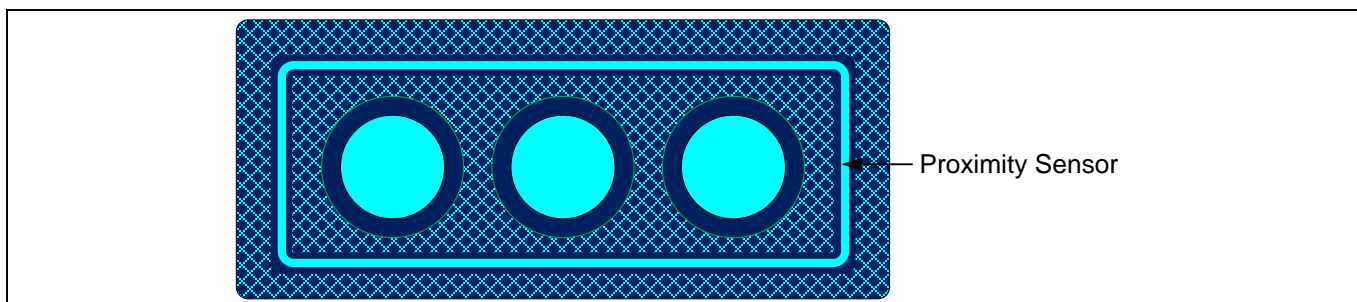


Figure 21 近接センサー

CAPSENSE™ 技術

他のセンサーと一緒に連結することで、近接センサーを実装することもできます。これは、ファームウェアを使用して1つの大型センサー内に複数のセンサーパッドを結合することで実現されます。この方法の欠点は高い寄生容量です。特定のデバイスでサポートされる最大寄生容量の詳細については、[コンポーネントデータシート/ミドルウェア文書](#)を参照してください。

近接センサーの詳細は、[AN92239 proximity sensing with CAPSENSE™](#)および [Getting started with CAPSENSE™ design guide](#) の近接センシングの節を参照してください。

2.5 耐液性

静電容量センシングは、家電、自動車、および産業応用などのさまざまな用途で使用されます。これらのアプリケーションは、霧、湿気、水、氷、湿度、またはその他の液体の存在下でも堅牢な静電容量センシングの動作を必要とします。静電容量センシングアプリケーションの設計では、誤ったタッチ検知または近接検出は、センサー面上の液膜か液滴の存在 (液体の導電性) に起因して起こることがあります。CSD センシング方式は、これらの要因による raw カウントの変化を補正でき、堅牢かつ信頼性の高い静電容量センシングアプリケーション動作を提供します。



Figure 22 洗濯機での耐液性のある CAPSENSE™ ベースのタッチ ユーザー インターフェース

- CAPSENSE™ センシング方式では、霧、湿気、湿度変化による raw カウントの変化を補正するために、センサーのベースラインを連続的に調整することで誤ったトリガーを防止します。
- 液体の流れによるセンサーの誤ったトリガーを防ぐためには、[Figure 23](#) に示すように [ガードセンサー](#) を実装する必要があります。[被駆動シールド信号およびシールド電極](#) を使用して、流れる液体の存在を検出し、ステータスを無視するか、液体の流れが存在する限り、残りのセンサーからの検知を停止できます。
- 液体の流れの許容範囲についても、相互容量センサーの周囲に実装できますが、ガードセンサー自体は単なる別の自己容量センサーであることに注意してください。PSoC™ デバイスを使用すると、このような自己容量センサーと相互容量センサーを同じ設計で一緒に実装できます。
- 自己容量センシング用の液滴による raw カウントの変化を補正するために、[Figure 23](#) に示すように [被駆動シールド信号およびシールド電極](#) を実装できます。シールド電極を実装すると、センサー表面に液滴が存在する場合でも、CAPSENSE™ が確実に機能し、センサーの ON/OFF ステータスを正しく報告します。相互容量検知用の液滴によるセンサーの誤ったトリガーを防ぐために、[相互容量センサーの耐液性のための自己容量センシングの使用](#) が説明するのと同じセンサーのセットで、[被駆動シールド信号およびシールド電極](#) による相互容量と自己容量の両方のセンシング方法を使用できます。

要約すると、アプリケーションで液滴に対する耐性が必要な場合は、[被駆動シールド信号およびシールド電極](#) を実装します。アプリケーションで液滴と一緒に液体を流すことに対する耐性が必要な場合

CAPSENSE™ 技術

は、Figure 23 に示すように、被駆動シールド信号およびシールド電極およびガードセンサーを実装してください。耐液性のレイアウトガイドラインセクションで説明されている回路図とレイアウトのガイドラインに従って、それぞれシールド電極とガードセンサーを構築してください。

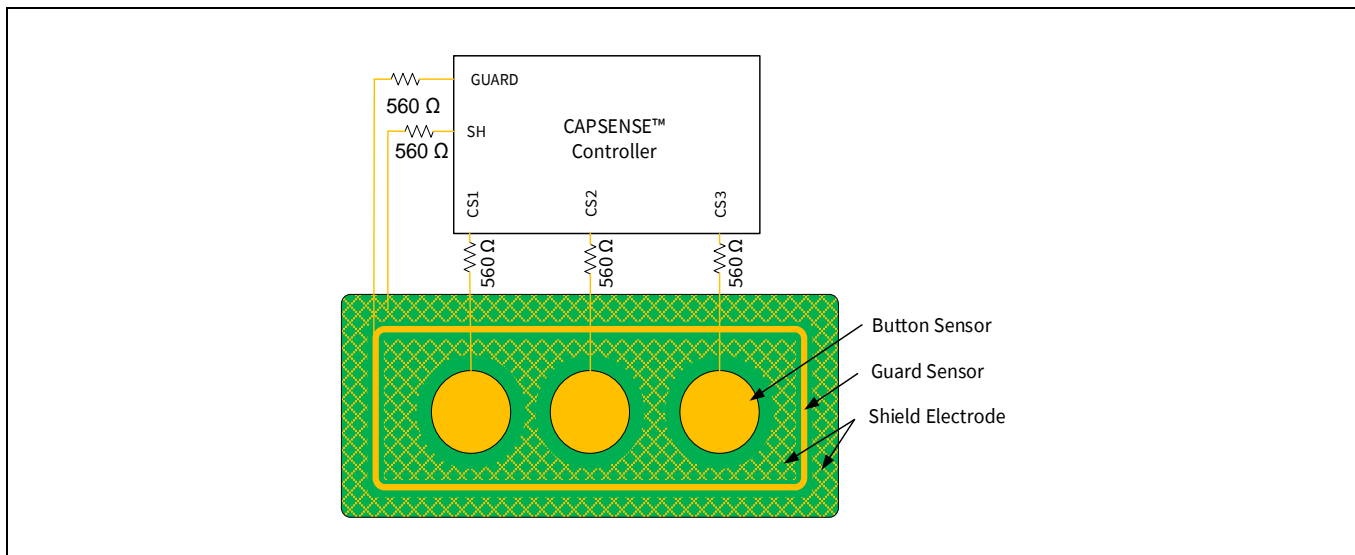


Figure 23 CAPSENSE™ コントローラーに接続したシールド電極 (SH) およびガードセンサー (GUARD)

2.5.1 自己容量センシングの耐液性

2.5.1.1 自己容量センサーに対する液滴と液体流の影響

CAPSENSE™ センサーに対する液体の影響を理解するために、Figure 24(a) に示す、センサーを取り囲んだハッチがグランドに接続されている CAPSENSE™ システムを見てください。ハッチを GND に接続することによって、センサーのノイズ耐性が改善されます。センサーの寄生容量は、Figure 24(b) では C_p と表記されます。

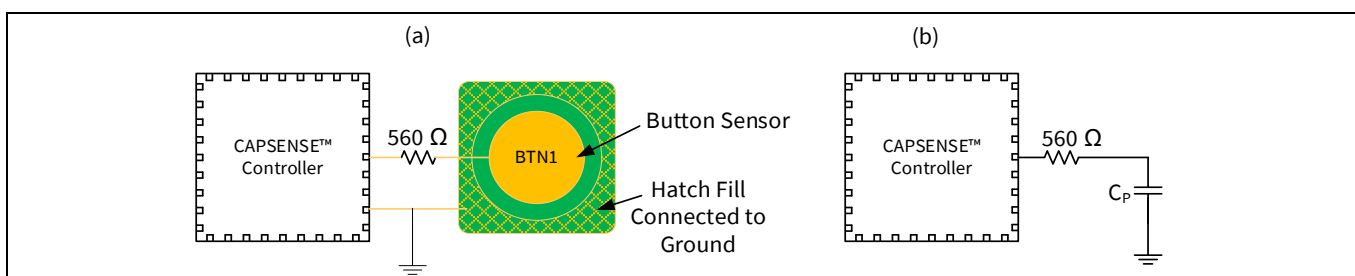


Figure 24 標準的 CAPSENSE™ システムのレイアウト

Figure 25 に示すように、液滴がセンサー面上に落下すると、その導電性により、磁力線がグランドに戻るために強い結合経路を提供します。その結果 C_p と並列に C_{LD} の静電容量を追加します。この追加静電容量は、GPIO セルの静電容量-電流変換器で説明したように、AMUX バスから更なる電荷を消費します。その結果、センサーの raw カウントが増加されます。いくつかの場合 (塩水またはミネラルを含む水の場合)、Figure 25 に示すように、液滴のセンサー面上への落下による raw カウント増加は指のタッチによる raw カウント増加に等しくなります。その場合、センサーは誤ったトリガーを発生させることがあります。

CAPSENSE™ 技術

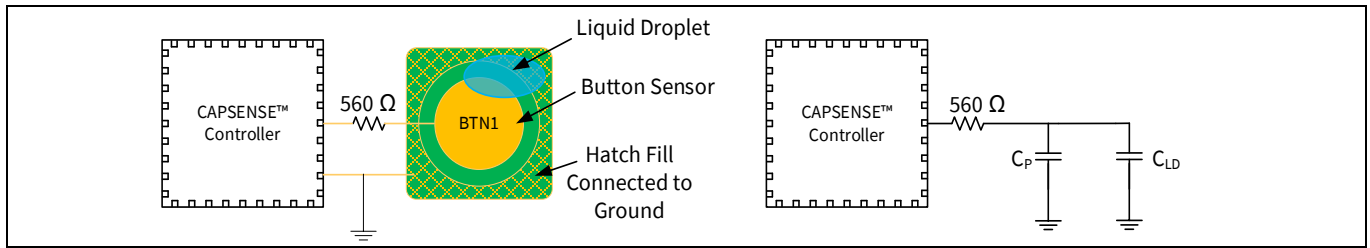


Figure 25 ハッチが GND に接続されているときの液滴により追加された静電容量

C_P = センサー寄生容量

C_{LD} = 液滴により追加された静電容量

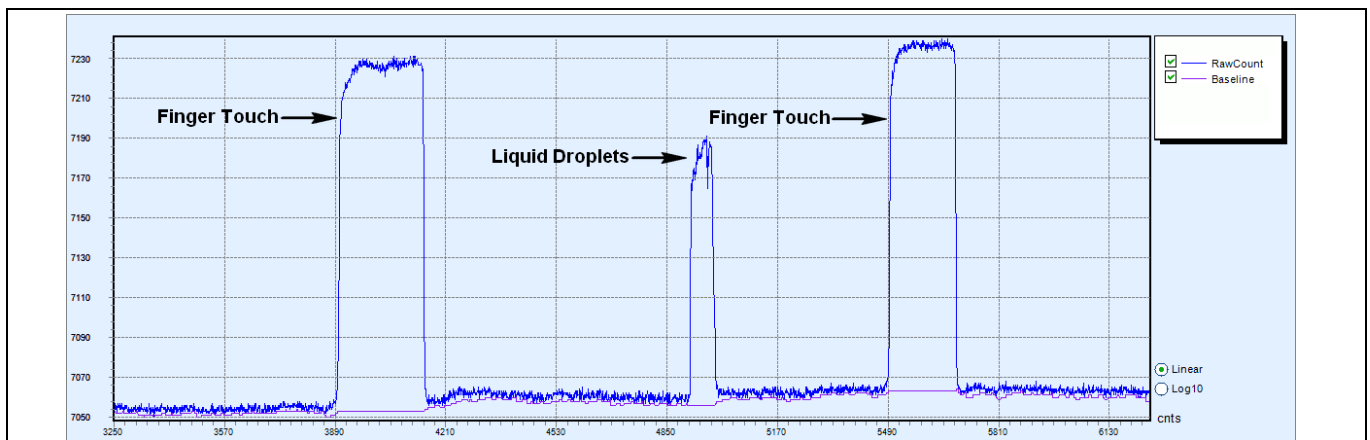


Figure 26 センサー周囲のハッチが GND に接続されているときの液滴の影響

液滴による CAPSENSE™ の回路に追加された静電容量の影響をなくさせるために、**被駆動シールド信号**でセンサー周囲のハッチを駆動する必要があります。

Figure 27 に示すように、センサー周囲のハッチを被駆動シールド信号に接続されたとき、および液滴がタッチインターフェースに当たったとき、液滴の両側の電圧は同電位に保たれます。そのため、液滴により追加された静電容量 (C_{LD}) は AMUX バスから更なる電流を引き出しません。よって、静電容量 C_{LD} の影響がなくなります。したがって、Figure 28 に示すように、液滴がセンサーに落下することによる raw カウントの増加は非常に小さいです。

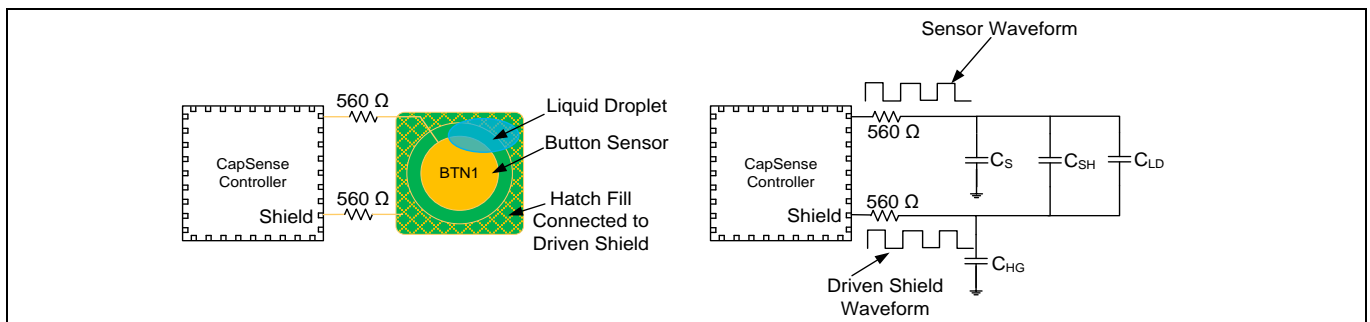


Figure 27 センサー周囲のハッチがシールドに接続されているときの液滴により増加した静電容量

C_S = センサー寄生容量

CAPSENSE™ 技術

C_{SH} = センサーとハッチ間の静電容量

C_{HG} = ハッチとグランド間の静電容量

C_{LD} = 液滴により追加された静電容量

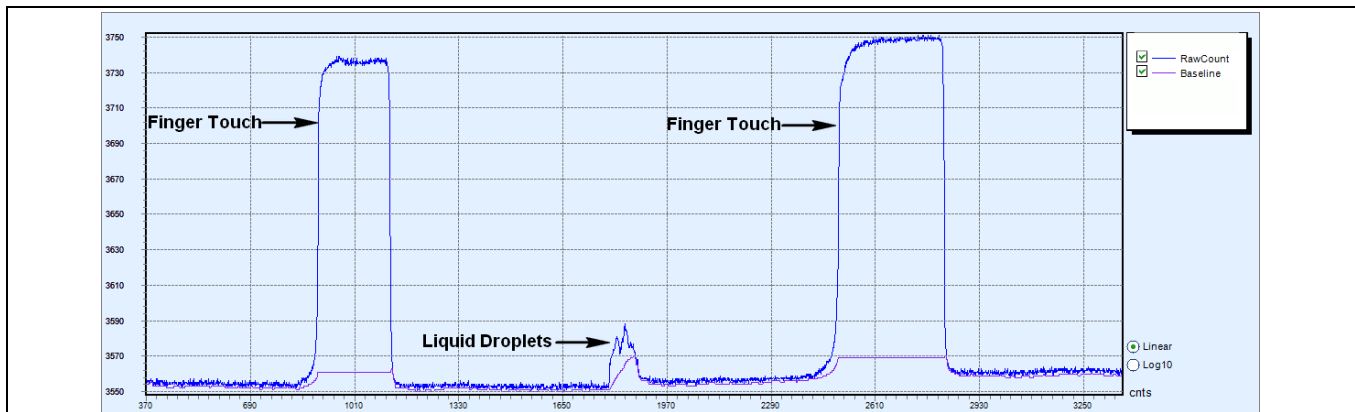


Figure 28 センサー周囲のハッチが被駆動シールドに接続されているときの液滴の影響

Figure 26 に、ハッチフィルがアースに接続される場合に、液体が存在する場合にセンサーが誤ってトリガーする方法を示します。ただし、すべての場合に同じことが当てはまるわけではないことに注意してください。たとえば、PCB 表面にエッチングされたセンサーよりも本質的に耐液性の高い **スプリングセンサー** の場合はあてはまりません。Figure 29 に示すように、液滴とハッチ充填の間のエアギャップが大きいため、PCB 上の液滴と接地されたハッチパターンとの間の静電容量 C_{LD} は、誤ったトリガーを引き起こさないように非常に低くなります。必要に応じて、PCB のハッチングパターンを被駆動シールド電極に接続して、 C_{LD} の影響をさらに無効にし、耐液性を向上させられます。

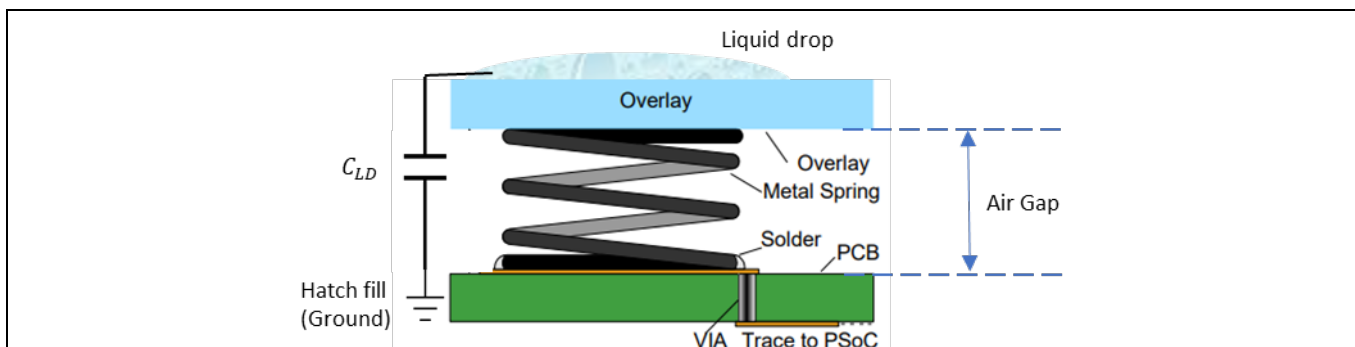


Figure 29 スプリングセンサーの液滴によって追加された容量

2.5.1.2 被駆動シールド信号およびシールド電極

Figure 30 に示すように、被駆動シールド信号は、センサースイッチング信号のバッファリングされたバージョンです。被駆動シールド信号は、センサースイッチング信号と同じ振幅、周波数、および位相を有します。センサー周囲のハッチは、被駆動シールド信号に接続されるとき、シールド電極と呼ばれます。

CAPSENSE™ 技術

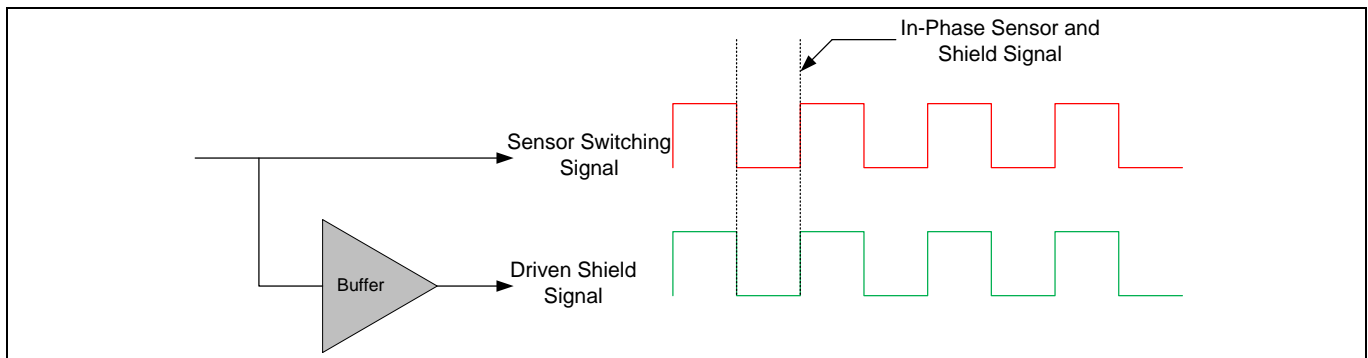


Figure 30 被駆動シールド信号

シールド電極は以下の目的で使用できます。

- **耐液性 CAPSENSE™ 設計の実装:** シールド電極は、[上記](#)で説明したように CAPSENSE™ 設計に耐液性を持たせるために有用です。
- **開放/グランド接続の導電性物体の存在下で近接距離センシングの向上:** 近接センサーと開放/グランド接続の導電性物体との間に配置されたシールド電極は、近接センシング距離に対するこれらの物体の影響を軽減し、遠い近接センシング距離を達成するために役立ちます。 [Getting started with CAPSENSE™ design guide](#) の「Proximity Sensing」セクションを参照してください。
- **センサーの寄生容量の低減:** CAPSENSE™ のセンサーが長い配線を持つ場合、センサー配線から周囲のグランドへのセンサー電界線の増加したカップリングにより、センサーの CP は非常に高いです。シールド電極を実装することで、グランドへの電界線のカップリングが削減されるため、センサーの CP は低下されます。

シールド電極のレイアウトガイドラインについては[シールド電極のレイアウトガイドライン](#)を参照してください。

2.5.1.3 ガードセンサー

連続する液体の流れがセンサー表面に存在すると、CAPSENSE™ のセンサーに大きい静電容量 (C_{ST}) を追加します。この静電容量は、 C_{LD} の数倍大きくなる可能性があります。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 raw カウントは、指のタッチと同じかそれより高くなります。このような状況では、ガードセンサーがセンサーの誤ったトリガーを防止するために役立ちます。

ガードセンサーは、[Figure 31](#) に示すように、PCB 上のすべてのセンサーを取り囲んでいる銅配線です。ガードセンサーはボタンセンサーと同様に、液体流の存在を検出するために使用されます。ガードセンサーがトリガーされると、ファームウェアはセンサーの誤ったトリガーを防ぐために、ガードセンサー以外のすべての他のセンサーのスキャンを無効にする必要があります。

Note: *ガードセンサーがトリガーされたときに、センサーがスキャンされないか、またはセンサーステータスが無視されます。したがって、タッチ面に液体の流れがある場合、タッチを検出できません。*

CAPSENSE™ 技術

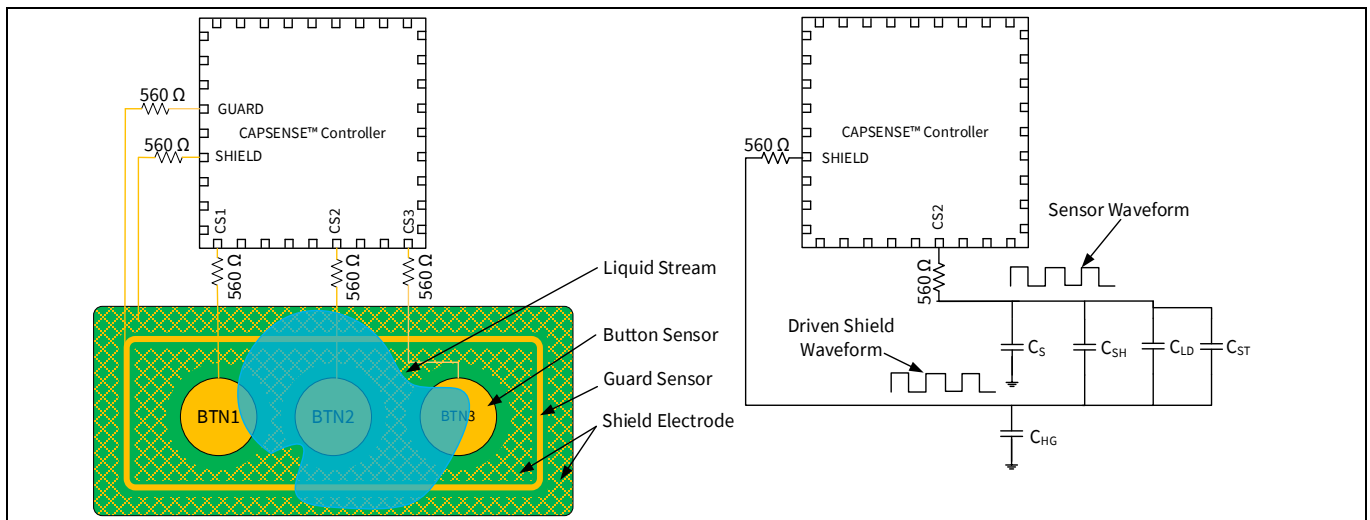


Figure 31 液体流がある場合の静電容量測定

ガードセンサーを実装するための PCB レイアウトガイドラインについては、[ガードセンサーのレイアウトガイドライン](#)を参照してください。

PCB にガードセンサーを実装するためのスペースがない場合は、ガードセンサー機能をファームウェアに実装できます。たとえば、次のように、さまざまなセンサーの ON/OFF ステータスを使用して、ユースケースに応じて液体の流れを検出できます。

- 液体の流れがある場合、1 度に複数のボタンセンサーがアクティブになります。デザインでマルチタッチセンシングが必要ない場合は、これを検出し、すべてのボタンセンサーのセンサーステータスを無視して、誤ったトリガーを防げます。
- スライダーで、ON になっているスライダーセグメントが隣接するセグメントでない場合は、スライダーセグメントのステータスをリセットするか、計算されたスライダーの重心値を無視できます。
- 同様に、独自のカスタムアルゴリズムを作成して、流れる液体の存在を検出し、タッチ面に液体が存在する間はセンサーのステータスを無視できます。

Note: センサーはスキャンされないか、ガードセンサーがトリガーされたときにセンサーステータスが無視されます。したがって、タッチ面に液体の流れがある場合、タッチを検出できません。

2.5.2 相互容量センシングの耐液性

2.5.2.1 相互容量センサーに対する液滴と液体の流れの影響

相互容量ボタンには、ノイズ耐性を向上させるために、センサーの周囲に接地されたハッチが埋め込まれていることがしばしばあります。接地されたハッチの一部を覆っている間に液滴がセンサーの上に落ちると、センサーに指を置いた場合と同様に、相互容量が減少します。[CAPSENSE™ CSX センシング方式 \(第 3 および第 4 世代\)](#)で説明されるように、および [Figure 32](#) に示すように、この相互容量の減少によって raw カウントが増加します。raw カウントの増加量は、液滴のサイズと特性によって異なります。

ただし、液滴が Tx および Rx 電極のみを覆い、接地されたハッチ上に広がらない場合、相互容量は増加します。これにより、[Figure 32](#) のように、この相互容量の減少によって raw カウントが減少します。この raw カウントの減少によって、[低ベースラインリセット](#)によるベースラインリセットが発生する可

CAPSENSE™ 技術

能性があります。液滴が除去されると、ベースラインが低い値のままである間、raw カウントが増加し、差信号が発生して、センサーが誤ってトリガーされる可能性があります。

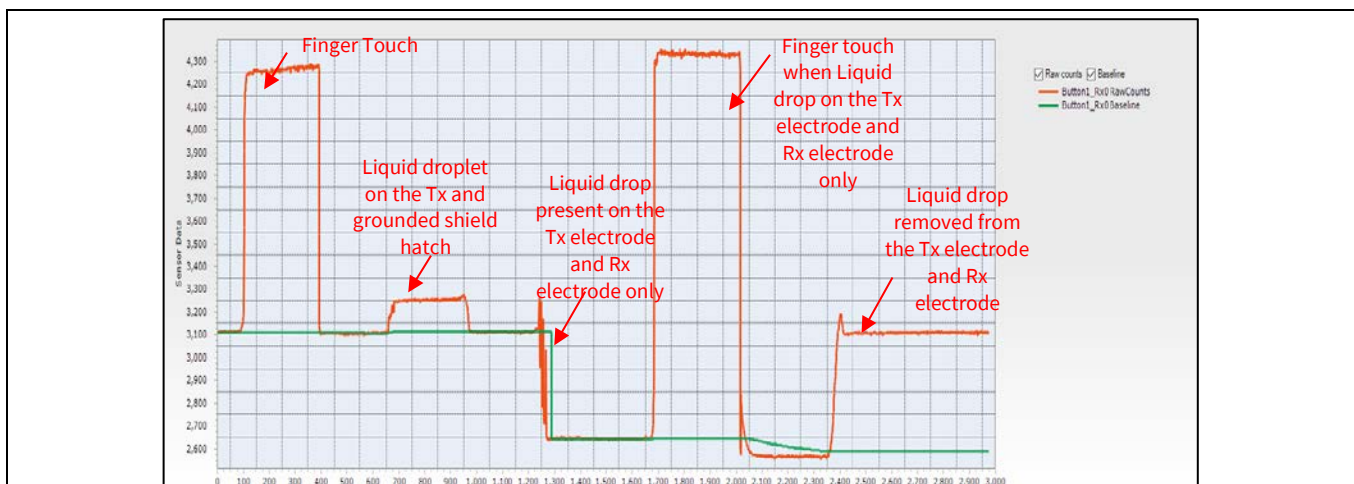


Figure 32 センサーの周りのハッチフィルがアースに接続される場合の CSX センサーに対する液滴の影響

2.5.2.2 相互容量センサーの耐液性のための自己容量センシングの使用

CAPSENSE™は、相互容量センサーの Tx ノードと Rx ノードの自己容量を検知します。この、CSD モードと CSX モードの両方を使用してセンサーをスキャンする機能を使用することによって、相互容量センサーに存在する液滴による誤ったトリガーを回避できます。CSD と CSX の両方の検出方法で相互容量ボタンを検出する方法を理解するためには、CAPSENSE™を使用したサンプルコード [PSoc™ 4 hybrid sensing using CAPSENSE™](#)を参照してください。

耐液性を実現するためには、CSD センス方式でセンサーの Rx 電極をスキャンする必要があります。Rx 電極を CSD センサーとしてスキャンするときは、シールド電極が有効になっていることを確認し、相互容量センサーの Tx ピンを被駆動シールド信号に接続します。以下に示すように、Rx 電極を CSD センサーとしてスキャンする `Capsense_ScanAllWidgets()` API 関数を呼び出す前に、低レベル API 関数 `Capsense_SetPinState()` を使用して相互容量センサーの Tx ピンをシールド電極に接続できます。

```
Capsense_SetPinState(Capsense_BUTTON1_WDGT_ID, Capsense_BUTTON1_TX0_ID, Capsense_SHIELD);

Capsense_ScanAllWidgets();
```

セクション 2.5.1 と 2.5.2 によって、それぞれ CSD ボタンと CSX ボタンに対する液滴の影響を説明しました。液滴に対する応答の違いを利用することで、ファームウェアロジックを作成して、耐液性の相互容量センサーを実現できます。CSD および CSX スキャン結果に対する液滴の存在の影響を Figure 33 に要約します。

CAPSENSE™ 技術

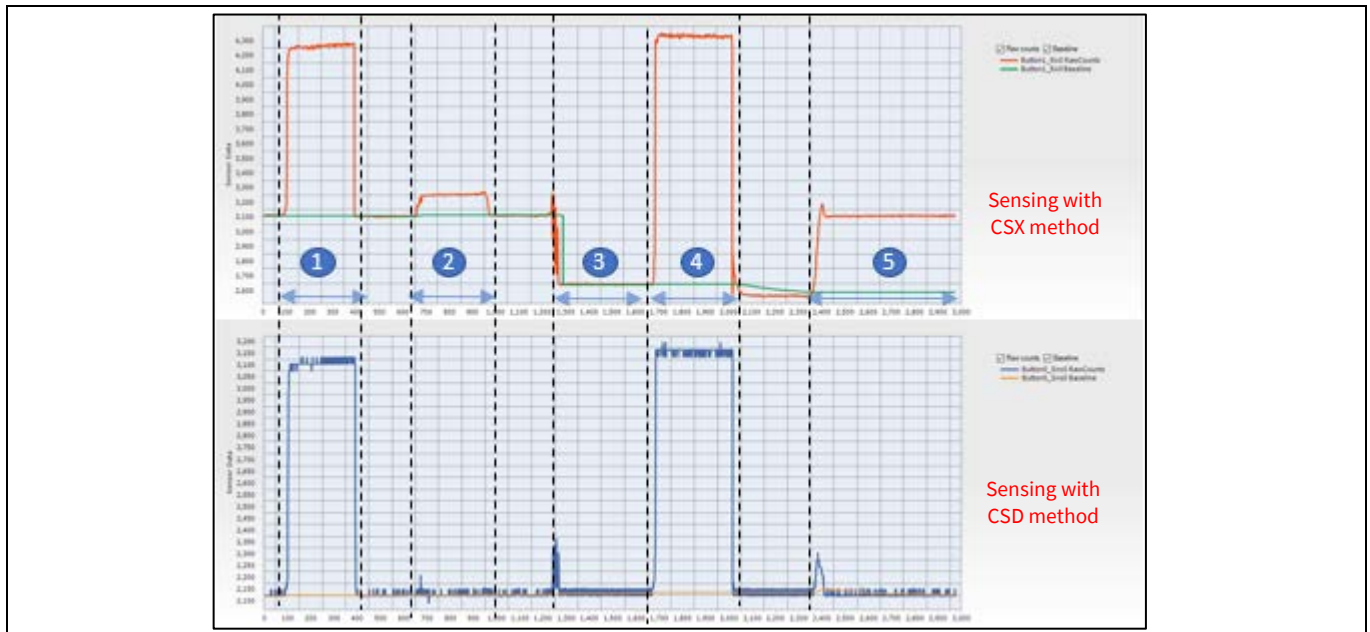


Figure 33 CSD および CSX 方式でスキャンされた CSX センサーパターンに対する水滴の影響

Figure 33 に、この方法を使用してスキャンした場合の、ハッチフィルで囲まれた CSX センサーパターンに対する水滴の影響を示します。Figure 33 の領域は、次のことを表します。

1. 指でのタッチ
2. Tx ラインと接地されたシールドハッチ上の液滴
3. Tx および Rx 電極にのみ存在する液滴
4. 液滴が Tx および Rx 電極のみにある場合の指の接触
5. Tx および Rx 電極から除去された液滴

Figure 33 に示すように、raw カウントの変更をファームウェアで使用して、CSX センサーのベースラインをリセットし、液滴の影響を無効にできます。ボタンのステータスは、領域 1、4 の場合はオン状態、その他の領域の場合はオフ状態である必要があります。さらに、CSX ボタンのベースラインは領域 3 と領域 5 で再初期化する必要があります。センサーのベースラインは、以下に示すように Capsense_InitializeWidgetBaseline() API 関数を使用してリセットできます。

```
Capsense_InitializeWidgetBaseline(Capsense_CSX_BUTTON_WDGT_ID);
```

この API の使用の詳細については、[コンポーネントデータシート/ミドルウェア文書](#)を参照してください。センサーのベースラインについては、[CAPSENSE™ソフトウェアパラメーターの選択](#)を参照してください。

2.5.3 耐液性性能への液体属性の影響

いくつかのアプリケーションでは、CAPSENSE™のシステムは、石鹼水、海水やミネラル水などのさまざまな液体の存在下で動作しなければなりません。そのようなアプリケーションでは、液滴による最悪ケースの信号を考慮してセンサーの CAPSENSE™パラメーターを調節することを推奨します。最悪ケースの条件をシミュレートするために、1 リットルの水に 40 グラムの調理塩 (NaCl) を溶かす塩水でセンサーの耐液性性能をテストすることを推奨します。石鹼水を用いて行ったテストの結果は石鹼水の影響が塩水の影響と同様であることを示します。そのため、チューニングを行って塩水を拒否する場合、CAPSENSE™システムは石鹼水の存在下でも動作できます。

CAPSENSE™ 技術

IH 調理器などの応用では、CAPSENSE™ タッチ面にお湯がこぼれることがあります。CAPSENSE™ 性能への液滴の温度の影響を判断するためには、異なる温度での液滴をセンサーに注いで、対応する raw カウントの変化を監視します。Figure 34 に示すように、実験結果は高温液滴の影響が室温での液滴の影響と同じであることを示します。これは、高温液滴がタッチ面に当たると直ちに室温に冷えるためです。お湯が継続的にセンサーに当たるため、オーバーレイの温度が上昇すると、温度上昇による raw カウントの増加は [ベースライン更新アルゴリズム](#) によって補正されます。よって、センサーのあらゆる誤ったトリガーを防止できます。

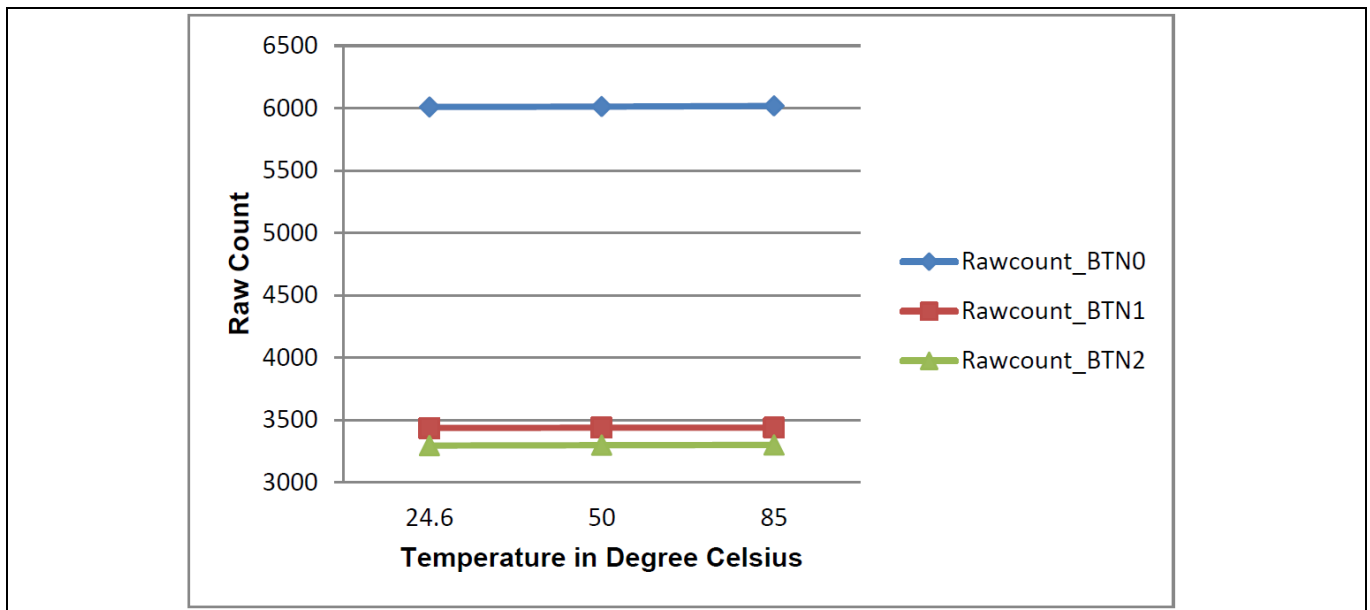


Figure 34 水の温度と raw カウントの変化

PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™

3 PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™

ここでは、CAPSENSE™ CSD および CSX を PSoC™ 4 および PSoC™ 6 MCU に実装する方法について説明します。CAPSENSE™ の基本原則を理解するためには、[静電容量タッチセンシング方式](#)を参照してください。PSoC™ デバイス アーキテクチャの基礎知識は本章を読むうえでの必要条件です。PSoC™ 4 を初めてお使いの場合は、[AN79953 – Getting started with PSoC™ 4](#) または [AN91267 – Getting started with PSoC™ 4 Bluetooth® LE](#) を参照してください。PSoC™ 6 MCU については [AN221774 – Getting started with PSoC™ 6 MCU](#) を参照してください。

コンポーネントの自動チューニング機能 (SmartSense) を使用する場合、本章を飛ばしてください。詳細は [CAPSENSE™ 性能のチューニング](#) を参照してください。

PSoC™ 4 デバイス ファミリは 3 つの異なる CAPSENSE™ アーキテクチャを持ちます。Table 2 は、第 3 世代、第 5 世代の CAPSENSE™ アーキテクチャの違いを説明しています。

3.1 PSoC™ 4 および PSoC™ 6 の CAPSENSE™ 世代

Table 2 に、CAPSENSE™ アーキテクチャの主な違いを示します。

Table 2 CSD と CSX の CAPSENSE™ アーキテクチャの比較

特長	第 3 世代 CAPSENSE™	第 4 世代 CAPSENSE™	第 5 世代 CAPSENSE™	改善の影響	条件
SNR	5:1	6.5:1	48:1	より高い SNR は、より良い感度を意味する。つまりより小さな信号を感知する能力。	$V_{DD} = 5V$; firmware filter なし; $C_p \sim 33 \text{ pF}$; $C_f = 0.1 \text{ pF}$
センシングモード	自己容量および相互容量モード	自己容量, 相互容量モードおよび ADC モード	自己容量および相互容量モード	-	-
センサー寄生容量範囲	5 pF ~ 45 pF	5 pF ~ 200 pF	2 pF ~ 200 pF	C_p 範囲が広いということは、PCB レイアウトの配線の柔軟性が高く、非常に短い/長いセンサートレースで、さまざまな PCB 材料 (FFC など) を検出できることを意味する。	-
必要とされる標準センス信号	100 fF	100 fF	15 fF for CSD-RM 10 fF for CSX-RM	より小さなセンス信号が必要であり、より厚いオーバーレイ、より高い近接範囲、より小さなセンサーサイズなどのサポートを意味する。	$V_{DD} = 5V$; firmware filter なし; $C_p \sim 33 \text{ pF}$; SNR = 5:1;
ノイズフロア (pk-pk)	-	-	500 aF for CSD-RM 100aF for CSX-RM	SNR と同じ。 より高い SNR またはより低いノイズフロアは、より小さな信号を感知する能力を意味する。	$V_{DD} = 5V$; $C_p \sim 33 \text{ pF}$; $C_M = 5 \text{ pF}$

PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™

特長	第 3 世代 CAPSENSE™	第 4 世代 CAPSENSE™	第 5 世代 CAPSENSE™	改善の影響	条件
対応するオーバーレイの厚さ	最大 5 mm	最大 5 mm	最大 18 mm	より厚いオーバーレイの設計をサポート。	10 mm CSD ボタン; アクリルオーバーレイ; SNR = 5:1; $C_p \sim 22$ pF;
リフレッシュレート	-	22 Hz	242 Hz	リフレッシュレートが速いと、大きなトラックパッドや長いスライダー、または単一のデバイスを備えた多数のボタンセンサーなどのアプリケーションで、高速なジェスチャーとタップの検出が可能。	7x5 CSX タッチパッド; アクリルオーバーレイ 3mm 厚; SNR = 10:1; 指サイズ = 8 mm;
CPU バンド幅要求	完全に CPU 駆動。センサーの初期化とシーケンスには CPU が必要。	40% シーケンサー ⁴ は、センサーの初期化、構成、スキャンを処理します。各センサーのシーケンスに必要な CPU。	7% 完全に自律的。	センシングのための CPU 使用率を削減し、CPU を解放して他の周辺機器操作を実行し、アプリケーションの中央コントローラーとして機能する。	10x8 CSX タッチパッド; スキャンクロック = 1MHz; サブ変換数 = 70; リフレッシュレート = 100Hz;
排出制御オプション	PRS	PRS, SSC	PRS, SSC	-	-
ノイズ耐性	センス電圧 (Vref)	1.2V	1.2V~2.8V	Rail to Rail	センス電圧が高いほど、ノイズ耐性が高くなる。
	ディファレンシャルセンシング	相互容量センシング	相互容量センシング	相互容量および自己容量センシング	ディファレンシャルセンシングは、 C_{MOD} を介して外部環境から誘導されるノイズをキャンセルする。
	V_{DD} ノイズ	Yes	Yes	No	V_{DD} ノイズは、第 5 世代 CAPSENSE™ の動作に最小限の影響しか与えない。

⁴ ハードウェアステートマシンは、CAPSENSE™ ブロックとセンサー スキャンを制御するロジックです。

PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™

特長		第 3 世代 CAPSENSE™	第 4 世代 CAPSENSE™	第 5 世代 CAPSENSE™	改善の影響	条件
	ズ 影響					
センス クロック 周波数	自己 容量	45 kHz ~ 6 MHz	45 kHz ~ 6 M Hz	45 kHz ~ 6 MHz	センスクロック周波数が 高いほど、低 Cp センサ ーのスキャンが高速にな る。これにより、より高 速なタップまたはジェス チャーをサポートする機 能、または特定のリフレ ッシュレートに対して、 より優れた耐性を実現す るために複数の firmware filter を実装する機能が提 供される。	-
	相互 容量	45 kHz ~ 300 k Hz	45 kHz ~ 3 M Hz	45 kHz ~ 6 MHz		
マルチ チャンネル サポート		No	No	Yes	‘n’チャンネルが使用されて いる場合、同じ数のセン サーに対して‘n’倍のスキ ャン速度を提供する。	-
シールド Cp		--	--	1.2nF	-	-
デバイス ファミリ		PSoC™ 4100/42 00 PSoC™ 4100M/4 200M PSoC™ 4100L/4 200L PSoC™ 4100BL/ 4200BL	PSoC™ 4000 PSoC™ 4000 S PSoC™ 4100 S PSoC™ 4100 S Plus PSoC™ 6	PSoC™ 4100S Max	-	-

3.2 CAPSENSE™ CSD センシング方式

Figure 35 に CSD センシング モードで CAPSENSE™ センサーをスキャンする CAPSENSE™ ブロックを示します。

PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™

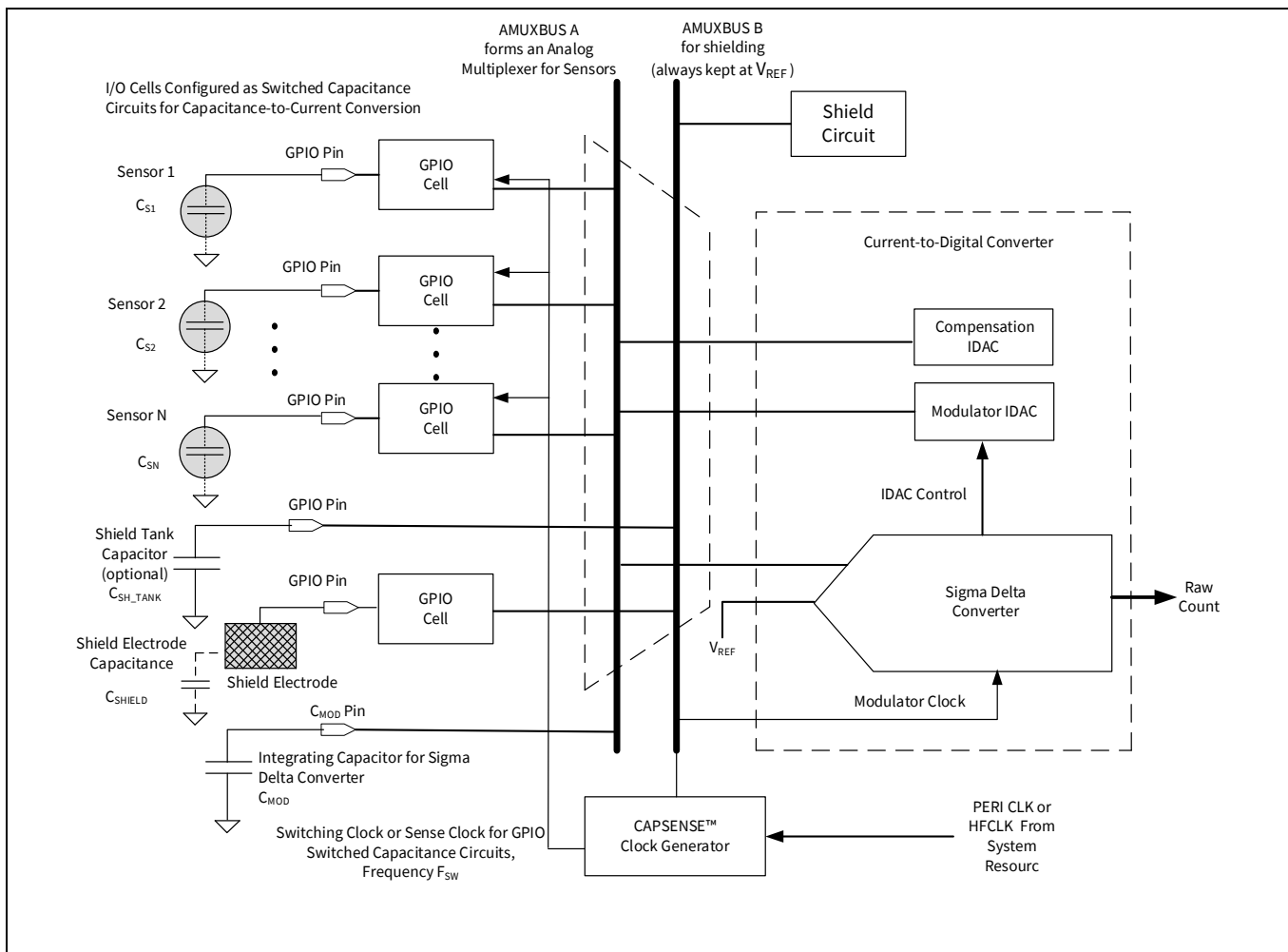


Figure 35 CAPSENSE™ CSD センシング

静電容量タッチセンシング方式で説明したように、このブロックはまずセンサー静電容量を等価な電流に変換することによって動作します。そしてアナログマルチプレクサは、複数の電流の1つを選択して、それを電流-デジタル変換器へ供給します。この電流-デジタル変換器は、特定の期間の変調 IDAC を制御するシグマ-デルタ変換器で構成されます。IDAC によってソースまたはシンクされる合計電流は、センサー容量によってシンクまたはソースされる合計電流と同じです。シグマ-デルタ変換器のデジタルカウント出力はセンサー静電容量のインジケータであり、raw カウントと呼ばれます。このブロックは IDAC ソース モードまたは IDAC シンク モードのいずれかで設定されます。IDAC ソース モードでは、IDAC は AMUXBUS に電流をソースし、GPIO セルは AMUXBUS から電流をシンクします。IDAC シンク モードでは、IDAC は AMUXBUS から電流をシンクし、GPIO セルは AMUXBUS に電流をソースします。

3.2.1 GPIO セルの静電容量-電流変換器

CAPSENSE™ CSD システムでは、GPIO セルは、センサーの容量を等価電流に変換するスイッチト キャパシタ回路として構成されます。Figure 36 に GPIO セル構造の概略図を示します。

PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™

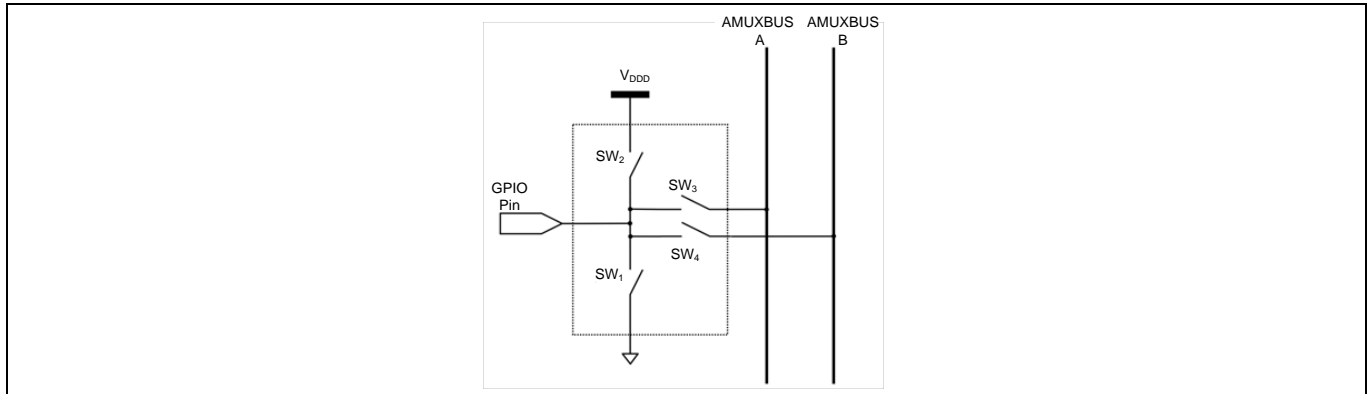


Figure 36 GPIO セル構造

PSoC™ 4 デバイスは2つのアナログマルチプレクサバスを持ちます。それらはCSDセンシング用のAMUXBUS A、CAPSENSE™ CSD シールド用のAMUXBUS Bです。GPIOスイッチトキャパシタ回路は2つの可能な構成があります。それらは、AMUXBUS Aへ電流をソースする構成、とAMUXBUS Aから電流をシンクする構成です。

3.2.2 IDAC ソース モード

Figure 37 に示すように、IDAC ソース モードでは、GPIOセルがスイッチトキャパシタ回路を介してAMUXBUS A から電流をシンクします。

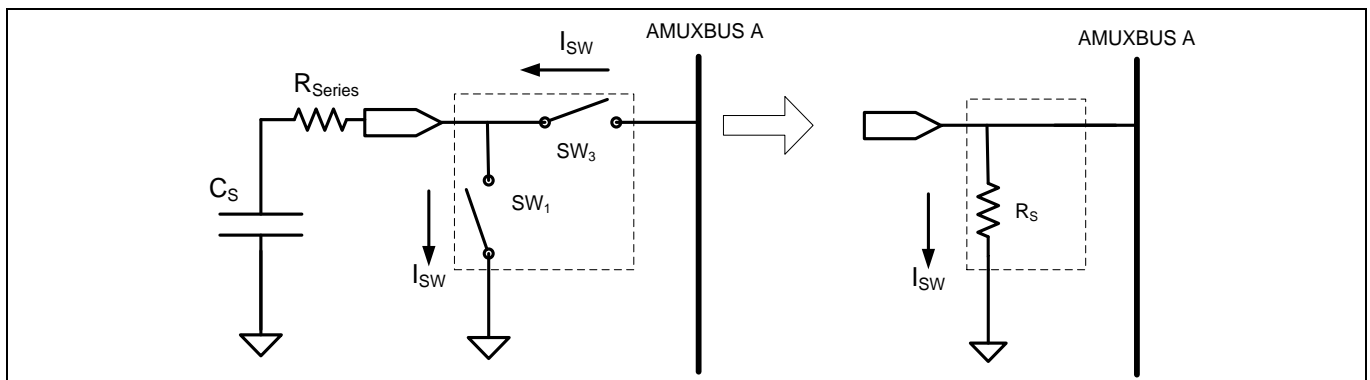


Figure 37 GPIO セルの AMUXBUS A からの電流シンク

Figure 38 に示すように、周波数 F_{SW} の位相が一致しない2つの重複しないクロックが、スイッチ SW_1 と SW_3 を制御します。Figure 37 に示すように、 SW_1 と SW_3 の連続切り替えにより、等価抵抗 R_s を形成します。

PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™

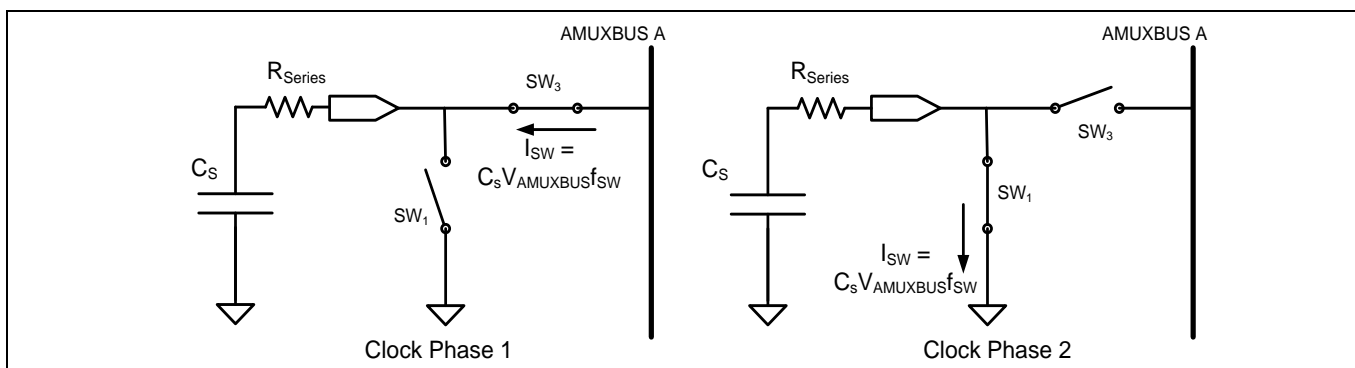


Figure 38 重複しない SW₁ と SW₃ スイッチ

スイッチが十分に低い周波数 F_{SW} で動作する場合、センサーを V_{REF} に完全に充電し、グラウンドに完全に放電するために時間 $(T_{SW}/2)$ は十分です。Figure 38 に示すように、等価抵抗 R_S の値は、式 5 によって与えられます。

式 5. センサーの等価抵抗

$$R_S = \frac{1}{C_S F_{SW}}$$

ここで、

C_S = センサーの静電容量

F_{SW} = センサー クロックの周波数

シグマ デルタ変換器は、 V_{REF} 定数で AMUXBUS A の電圧を維持します (このプロセスはシグマ デルタ変換器で説明されています)。Figure 39 は C_S 両端における電圧波形を示します。

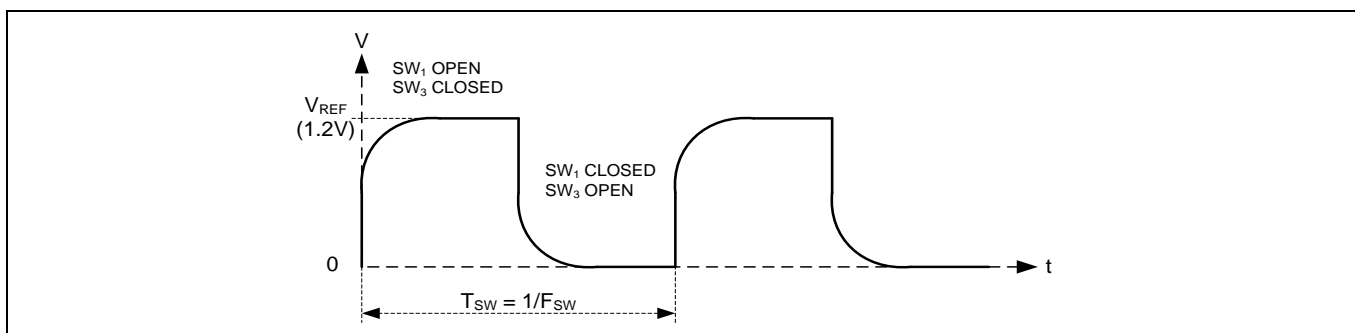


Figure 39 センサー静電容量の両端における電圧

式 6 により AMUXBUS A から供給される平均電流の値が得られます。

式 6. CAPSENSE™ センサーを介して AMUXBUS A から GPIO にシンクされる平均電流 (I_{CS})

$$I_{CS} = C_S F_{SW} V_{REF}$$

3.2.3 IDAC シンク モード

IDAC シンク モードでは、Figure 40 に示すように、GPIO セルはスイッチト キャパシタ回路を介して AMUXBUS A に電流をソースします。Figure 41 に、センサー静電容量の両端における電圧波形を示します。

PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™

IDAC ソース モードに比べて、このモードが直接 V_{DD} を通じて AMUXBUS A を充電するため、電源ノイズの影響を受けやすくなります。したがって、LDO または非常に安定した静かな V_{DD} とともにこのモードを使用することを推奨します。

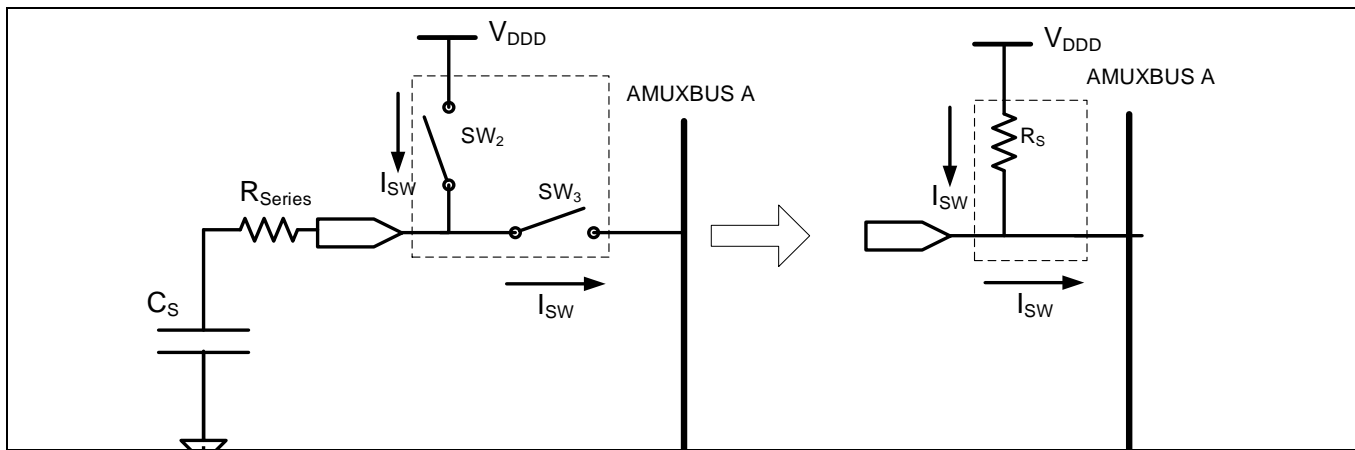


Figure 40 GPIO セルの AMUXBUS A への電流ソース

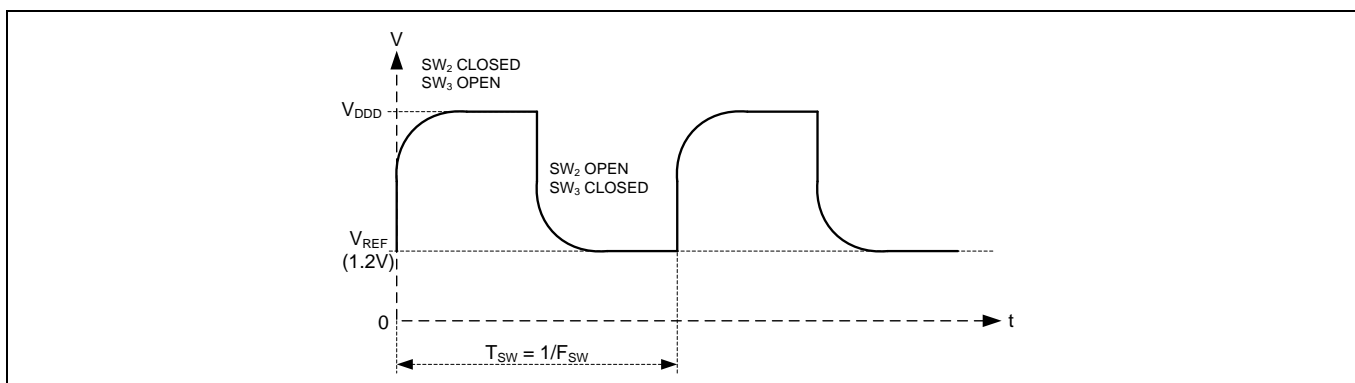


Figure 41 センサー静電容量の両端における電圧

式 7 により AMUXBUS A に供給される平均電流の値が得られます。

式 7. CAPSENSE™ センサーを介して GPIO から AMUXBUS A にソースされる平均電流 (I_{CS})

$$I_{CS} = C_S F_{SW} (V_{DD} - V_{REF})$$

3.2.4 CAPSENSE™ クロック ジェネレータ

Figure 35 に示すように、CAPSENSE™ クロックジェネレータブロックは、PSoC™ デバイスファミリに応じて、高周波数システム リソースクロック (HFCLK) またはペリフェラルクロック (PERI) から、センスクロック F_{SW} および変調クロック F_{MOD} を生成します。

3.2.4.1 センスクロック

センスクロック (スイッチングクロックとも呼ばれる) は、GPIO セルの静電容量-電流変換器のために、GPIO セルのスイッチトキャパシタ回路に供給する非重複クロックを駆動します。

このクロック出力は 3 つのオプションがあります。直接、8 ビット PRS および 12 ビット PRS です。一部の PSoC™ 4 および PSoC™ 6 MCU 製品は、追加のスペクトラム拡散クロック (SSC) モードもサポート

PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™

します。PSoC™ デバイスでサポートされているモードの詳細については、[コンポーネントデータシート / ミドルウェア文書](#)を参照してください。

直接クロックは、定周波数センスクロックソースです。このオプションを選択すると、センサーピンは、CAPSENSE™ コンポーネント構成ウィンドウで指定された周波数の一定周波数クロックで切り替わります。

PRS クロックは、センスクロックが 8 ビットまたは 12 ビットの PRS を生成できる PRS ブロックから駆動されることを意味します。PRS クロックを使用すると、PRS を使用して入力クロックを分周することにより、センスクロック周波数が広い周波数範囲に広がります。

SSCx は、センスクロック周波数も拡散します。優れたノイズ耐性を提供し、放射電磁放射を低減します。

クロックソースと周波数選択のガイドラインの詳細については、[ハードウェアパラメーターの手動調整](#)を参照してください。

3.2.4.2 変調クロック

変調クロックは、[シグマデルタ変換器](#)によって使用されます。このクロックは、[式 8](#) および [式 9](#) に基づいてセンサーのスキャン時間を決定します。

式 8. センサー スキャン時間

$$\text{センサースキャン時間} = \text{ハードウェアスキャン時間} + \text{センサー初期化時間}$$

式 9. ハードウェア スキャン時間

$$\text{ハードウェアスキャン時間} = \frac{(2^{\text{分解能}} - 1)}{\text{変調クロック周波数}}$$

ここで、

分解能 = [スキャン分解能](#)

センサー初期化時間 = センサーが内部レジスタに書き込み、スキャンを開始するためにかかる時間です。

3.2.5 シグマデルタ変換器

シグマデルタ変換器は入力電流を対応するデジタルカウントに変換します。この変換器は、[Figure 35](#) に示すように、シグマデルタ変換器と、変調 IDAC および補正 IDAC と呼ばれる 2 つの電流ソース/シンクのデジタル-アナログ変換器 (IDAC) で構成されます。

シグマデルタ変換器は、[Figure 35](#) に示すように、変調コンデンサ C_{MOD} と呼ばれる外部に積分コンデンサを使用します。シグマデルタ変換器は、 C_{MOD} の両端の小さな電圧変動に対応して変調 IDAC 電流を ON または OFF に切り替えて、 C_{MOD} 電圧を V_{REF} に維持することにより、変調 IDAC 電流を制御します。 C_{MOD} の推奨値を [Table 35](#) に示します。

シグマデルタ変換器は、IDAC ソース モードまたは IDAC シンク モードのいずれかで動作できます。

- **IDAC ソース モード:** このモードでは、[GPIO セルの静電容量・電流変換器](#)は、CMOD から AMUXBUS A を介して電流をシンクし、IDAC は AMUXBUS A に電流を供給して電圧のバランスを取ります。
- **IDAC シンク モード:** このモードでは、[GPIO セルの静電容量・電流変換器](#)は、CMOD から AMUXBUS A に電流を供給し、IDAC が AMUXBUS A を介して電流をシンクして電圧のバランスを取ります。

PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™

上記の両方のモードで、シグマデルタ変換器はシングル IDAC モードまたはデュアル IDAC モードのいずれかで動作できます。

- シングル IDAC モードでは、変調 IDAC はシグマデルタ変調器で制御され、補正 IDAC は常に OFF です。
- デュアル IDAC モードでは、変調 IDAC はシグマデルタ変調器で制御され、補正 IDAC は常に ON です。

シングル IDAC モードでは、'N' がシグマデルタ変換器の分解能であり、 I_{MOD} が変調 IDAC 電流の値である場合、IDAC ソースモードでの raw カウントの概算値は **式 10** で与えられます。

式 10. シングル IDAC ソースの raw カウント

$$\text{raw カウント} = (2^N - 1) \frac{V_{REF} F_{SW}}{I_{MOD}} C_S$$

同様に、IDAC シンクモードでの raw カウントの概算値は **式 11** で与えられます。

式 11. シングル IDAC シンクの raw カウント

$$\text{raw カウント} = (2^N - 1) \frac{(V_{DD} - V_{REF}) F_{SW}}{I_{MOD}} C_S$$

どちらの場合も、raw カウントはセンサーの静電容量 C_S に比例します。次に、raw カウントが CAPSENSE™ CSD コンポーネントファームウェアによって処理され、タッチが検出されます。 I_{MOD} 、 I_{COMP} 、および F_{SW} などのハードウェアパラメータとソフトウェアパラメータは、信頼性の高いタッチ検出のために最適な値に調整する必要があります。チューニングの詳細については、**CAPSENSE™性能のチューニング**を参照してください。

デュアル IDAC モードでは、補正 IDAC は常にオンです。 I_{COMP} が補正 IDAC 電流になると、IDAC ソースモードでの raw カウントを計算する式は **式 12** で計算されます。

式 12. デュアル IDAC ソースの raw カウント

$$\text{raw カウント} = (2^N - 1) \frac{V_{REF} F_{SW}}{I_{MOD}} C_S - (2^N - 1) \frac{I_{COMP}}{I_{MOD}}$$

IDAC シンクモードでの raw カウントは **式 13** で計算されます。

式 13. デュアル IDAC シンクの raw カウント

$$\text{raw カウント} = (2^N - 1) \frac{(V_{DD} - V_{REF}) F_{SW}}{I_{MOD}} C_S - (2^N - 1) \frac{I_{COMP}}{I_{MOD}}$$

raw カウントの値は常に正であることを注意してください。IDAC シンクモードに対して I_{COMP} が $(V_{DD} - V_{REF}) C_S F_{SW}$ より小さい、IDAC ソースモードに対して I_{COMP} が $C_S F_{SW} V_{REF}$ より小さいことを確認する必要があります。**式 13** は $I_{COMP} > V_{REF} C_S F_{SW}$ の場合は該当せず、**式 12** は $I_{COMP} > (V_{DD} - V_{REF}) C_S F_{SW}$ の場合は該当しません。これらの場合において、raw カウントはゼロになります。

CAPSENSE™ コンポーネントパラメータと上記の式に示されたパラメータの関係を **Table 3** に示します。

PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™

Table 3 CAPSENSE™ raw カウントと CAPSENSE™ CSD ハードウェアパラメーターの関係

Sl. No.	パラメーター	説明	備考
1	N	スキャン分解能	スキャン分解能は 6 ビット～16 ビットに設定できます。詳細については コンポーネント データシート/ミドルウェア 文書 を参照してください。
2	V _{REF}	該当なし	V _{REF} 値は 1.2V であるか、PSoC™ デバイスファミリに応じて 0.6V～V _{DDA} -0.6V の間で設定可能です。詳細については コンポーネント データシート/ミドルウェア 文書 を参照してください。
3	F _{SW}	センス クロック 周波数	センス クロック 周波数とセンス クロック ソースはセンサーのスイッチング周波数を決定します。詳細については センス クロック を参照してください。
		センス クロック ソース	
4	I _{MOD}	変調 IDAC	I _{MOD} =変調 IDAC 電流
5	I _{COMP}	補正 IDAC	I _{COMP} =補正 IDAC 電流
6	V _{DD}	該当なし	このパラメーターはデバイスの電源電圧です。
7	C _s	該当なし	このパラメーターはセンサーの寄生容量です。
8	該当なし	変調クロック周波数	変調クロック分周器は raw カウントの式に影響しません。詳細については 変調クロック を参照してください

3.2.6 アナログ マルチプレクサ (AMUX)

シグマデルタ変換器は 1 度に 1 個のセンサーをスキャンします。Figure 35 に示すように、アナログ マルチプレクサは複数の GPIO セルの 1 つを選択し、それをシグマデルタ変換器の入力に接続します。AMUXBUS A と GPIO セルスイッチ (Figure 40 の SW₃ を参照してください) は、このアナログ マルチプレクサを形成します。AMUXBUS A は CAPSENSE™ をサポートするすべての GPIO に接続します。CAPSENSE™ をサポートするポートピンの一覧については、該当の[デバイス データシート](#)を参照してください。また、AMUXBUS A は積分コンデンサ C_{MOD} をシグマデルタ変換器回路に接続します。AMUXBUS B はシールドに使用され、シールドが有効の場合は V_{REF} に保持されます。

3.2.7 CAPSENSE™ CSD シールド

PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™ は耐液性と近接センシング用のシールド電極に対応します。CAPSENSE™ は、センサーおよびシールド電極間の電位差を無効にするために、センサースイッチング信号のレプリカ信号でシールド電極を駆動するシールド回路があります。これが耐液性にどのように役立つかについての詳細は、[被駆動シールド信号およびシールド電極および自己容量センサーに対する液滴と液体流の影響](#)を参照してください。

センシング回路では、シグマデルタ変換器は AMUXBUS A を V_{REF} に保ちます ([シグマデルタ変換器](#)を参照してください)。GPIO セルは、AMUXBUS A と電源レール (設定によって、V_{DD} またはグラウンド) 間の[センサーを切り替える](#)ことでセンサー波形を生成します。シールド回路は同じ方法で動作します。AMUXBUS B は常に V_{REF} に保たれます。GPIO セルは AMUXBUS B と電源レール (センサーの設定と同じように、V_{DD} またはグラウンド) 間でシールドを切り替えます。このプロセスはシールド電極上にセンサースイッチング波形のレプリカを生成します。

寄生容量の大きい大きなシールド層の場合は、外部コンデンサ (Csh tank コンデンサ) を使用してシールド電極ドライバの駆動能力を高めます。

PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™

3.3 CAPSENSE™ CSX センシング方式 (第 3 および第 4 世代)

Figure 42 に、CSX センシング回路の簡略図を示します。実装では、CSD HW の以下のハードウェアサブブロックを使用します。

- 8 ビット IDAC とシグマ デルタ変換器
- AMUXBUS A
- Tx クロックおよび変調器クロック用の CAPSENSE™ クロックジェネレーター
- Tx および Rx 電極と外部キャップ用の V_{REF} とポートピン
- 2 つの外部コンデンサ (CINTA および CINTB)。(これらのコンデンサの推奨値については、Table 35 を参照してください)。

Note: PSoC™ 4100 は CSX センシング方式をサポートしていません。

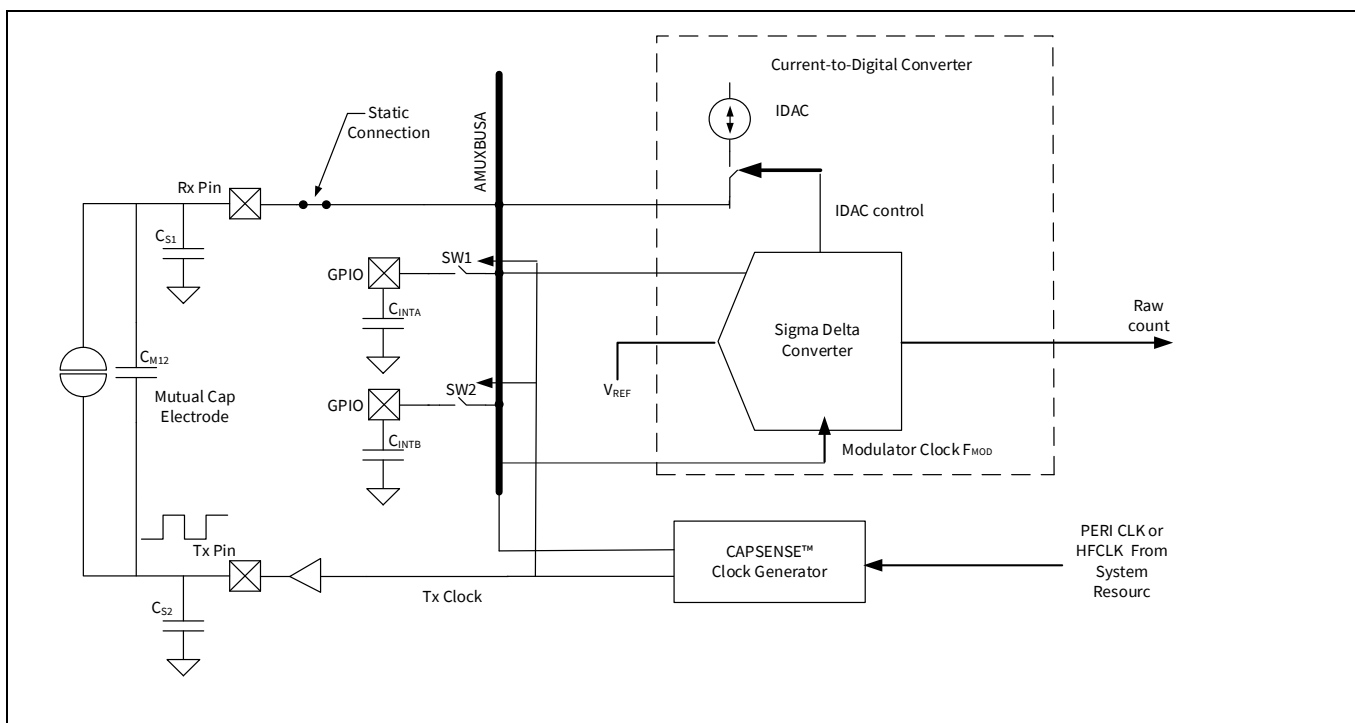


Figure 42 CAPSENSE™ CSX センシング方式の構成

CSX センシング方式では、Figure 42 に示すように、Tx 電極と Rx 電極間の相互容量を測定します。Tx 電極は、VDDIO (または指定された部品番号で VDDIO が使用できない場合は VDDD) とグラウンドの間で切り替わるデジタル波形 (Tx クロック) によって励起されます。Rx 電極は AMUXBUS A に静的に接続されます。CSX 方式では、 C_{INTA} と C_{INTB} の 2 つの外部積分コンデンサが必要です。これらのコンデンサの値を Table 35 に示します。

PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™

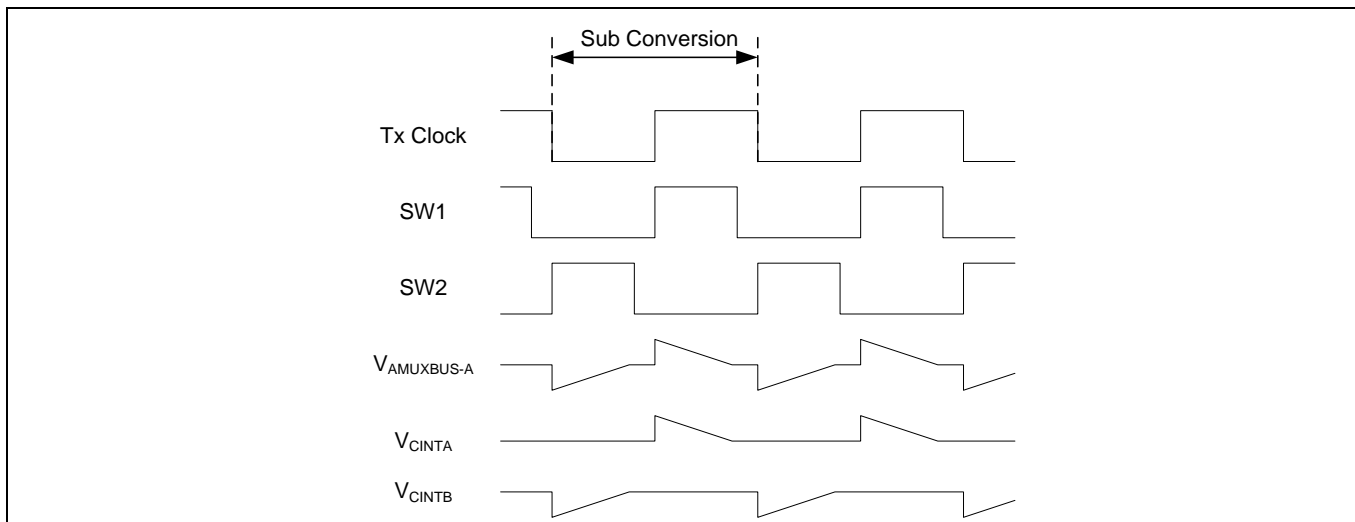


Figure 43 CSX センシング波形

Figure 43 に Tx 電極および C_{INTA} と C_{INTB} コンデンサの波形を示します。サンプリング (サンプルを生成するプロセス) は両方の外部コンデンサの電圧を V_{REF} に初期化し、一連のサブ変換を実行することでファームウェアによって開始します。サブ変換は、Tx クロックサイクル内で実行された変換をカウントするための静電容量です。サンプル内のすべてのサブ変換結果の合計は、「raw カウント」と呼ばれます。

サブ変換の間に、SW1 および SW2 スイッチの両方は Tx クロックと同相で動作します。Tx クロックの立ち上がりエッジで、SW1 が閉じ (このときに SW2 が開いている)、充電電流が Tx 電極から Rx 電極に流れます。この充電は C_{INTA} コンデンサに統合され、C_{INTA} の電圧を増加します。IDAC は元の電圧 V_{REF} に C_{INTA} コンデンサを放電するためにシンクモードで設定されます。

両方のサイクルで Tx 電極と Rx 電極間で転送される電荷は、電極間の相互容量 C_M に比例します。シグマデルタ変換器は、外部コンデンサを充電または放電するために IDAC を制御し、サブ変換の変調器クロックサイクルの観点から充電および放電時間を測定します。CSX スキャン中に複数のサブ変換が実行され、各サブ変換の結果が累積されて、センサーの「raw カウント」が生成されます。

変調クロックは Tx クロック サイクル以内の外部コンデンサを充電/放電する時間を測定するために使用します。そのため、変調クロック周波数はいつも Tx クロック周波数を超える必要があります。変調クロック周波数は高いほど精度が良くなります。正常動作のためには、IDAC 電流は 1 つの Tx クロック サイクル以内に C_{INTA} および C_{INTB} コンデンサを充電/放電できるように設定する必要があります。

CAPSENSE™ コンポーネント/ミドルウェアは IDAC を自動的に調整するオプションを提供します。このオプションを有効にすることを推奨します。

式 14. 相互容量センシングでの raw カウントの関係

$$\text{Rawcount}_{\text{Counter}} = \frac{2 V_{\text{TX}} F_{\text{TX}} C_{\text{M}} \text{MaxCount}}{\text{IDAC}}$$

$$\text{MaxCount} = \frac{F_{\text{Mod}} N_{\text{Sub}}}{F_{\text{TX}}}$$

ここで、

IDAC = IDAC 電流

C_M = Tx 電極と Rx 電極の間の相互容量

V_{TX} = Tx 信号の振幅

PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™

F_{TX} = Tx クロック周波数

F_{Mod} = 変調クロック周波数

N_{Sub} = サブ変換の数

CSX ボタンに指を置くと、Rx 電極と Tx 電極間の相互容量が減少し、raw カウントが減少します。ハードウェアからの raw カウントのこの減少は、CAPSENSE™ コンポーネントによって反転され、指で触れた場合の CSD の raw カウントの変化と同様になります。最終的に得られる逆生カウントは、式 15 で与えられます。

式 15. Rawcount_{Component} を決定する式

$$Rawcount_{Component} = MaxCount - Rawcount_{Counter}$$

CSX ハードウェアパラメーターの詳細については、CSX センシング方式 (第 3 および第 4 世代) を参照してください。

3.4 CAPSENSE™ CSD-RM センシング方式 (第 5 世代)

ここでは、第 5 世代 CAPSENSE™ (マルチセンスコンバーター (MSC) と呼ばれる) デバイスに実装されている CSD-RM アーキテクチャの概要を説明します。主な機能には、レシオメトリックセンシング、基準電圧を必要としない差動動作モード、ノイズ性能を向上させる電流 DAC (IDAC) の代わりにコンデンサ DAC (CDAC) を使用することが含まれます。

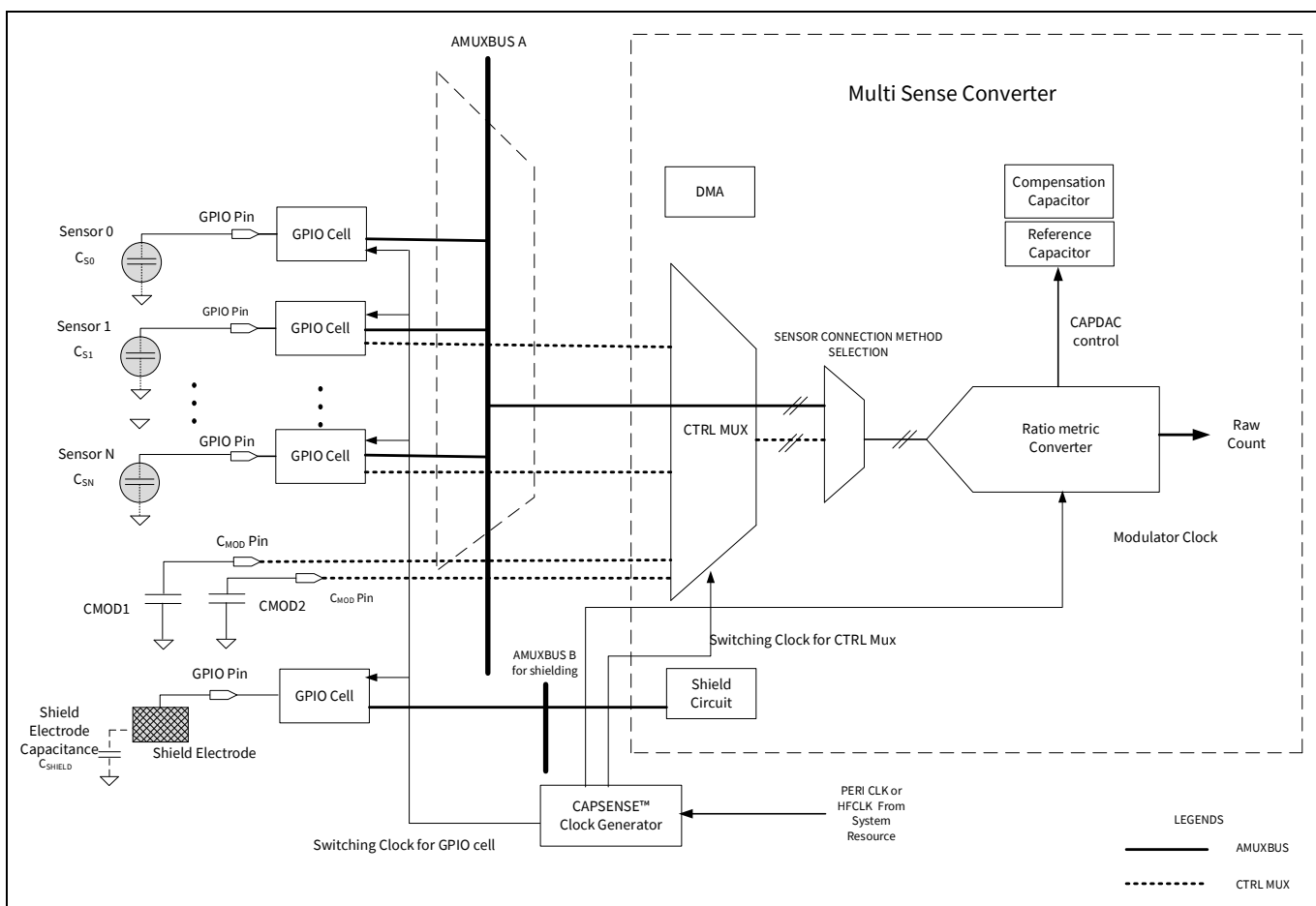


Figure 44 CAPSENSE™ CSD-RM (第 5 世代)

PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™

3.4.1 GPIO セルの容量から電荷へのコンバーター

セクション 3.2.1 では、GPIO セルの構成について説明します。第 5 世代のアーキテクチャでは、センサーは AMUX (以前と同様) または自律スキャンをサポートする新しい制御 MUX マトリックス (サポートされるピンの数に制限があります) のいずれかに接続されます。GPIO セルは、センサーの静電容量を同等の電荷移動に変換するスイッチトキャパシタ回路として構成されます。Figure 45 に、GPIO セルの構造を示します。

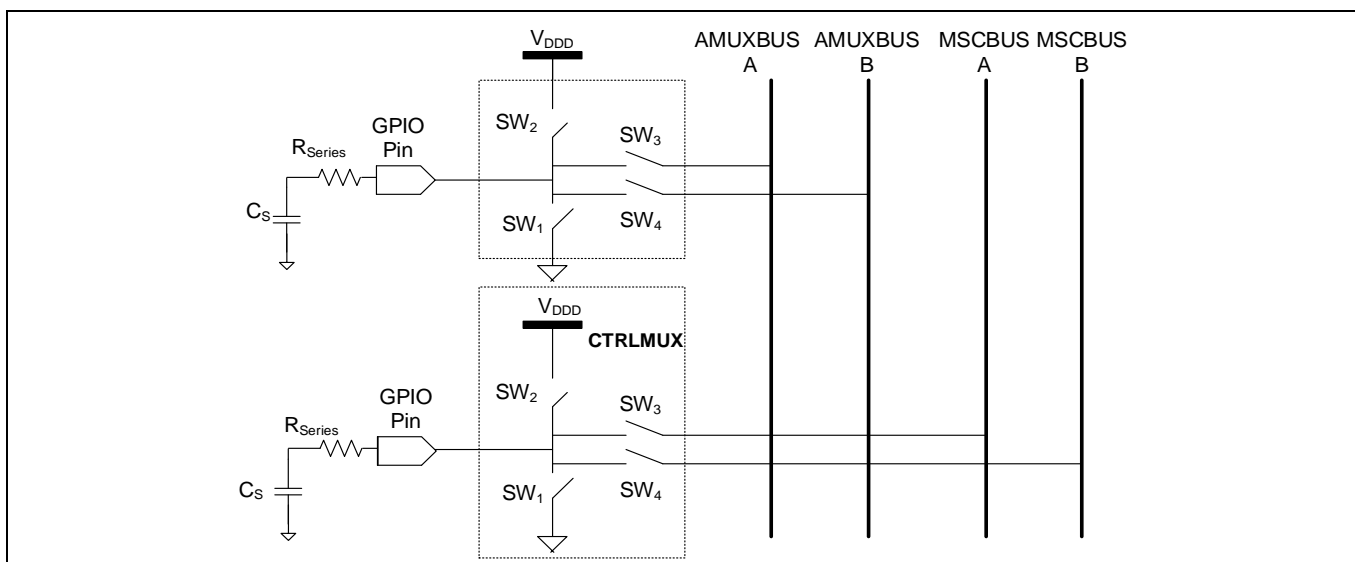


Figure 45 GPIO セル構造

Figure 46 に示すように、周波数 F_{SW} の 4 つの重複しない位相のずれたクロックが、スイッチ (SW1, SW2, SW3, および SW4) を制御します。

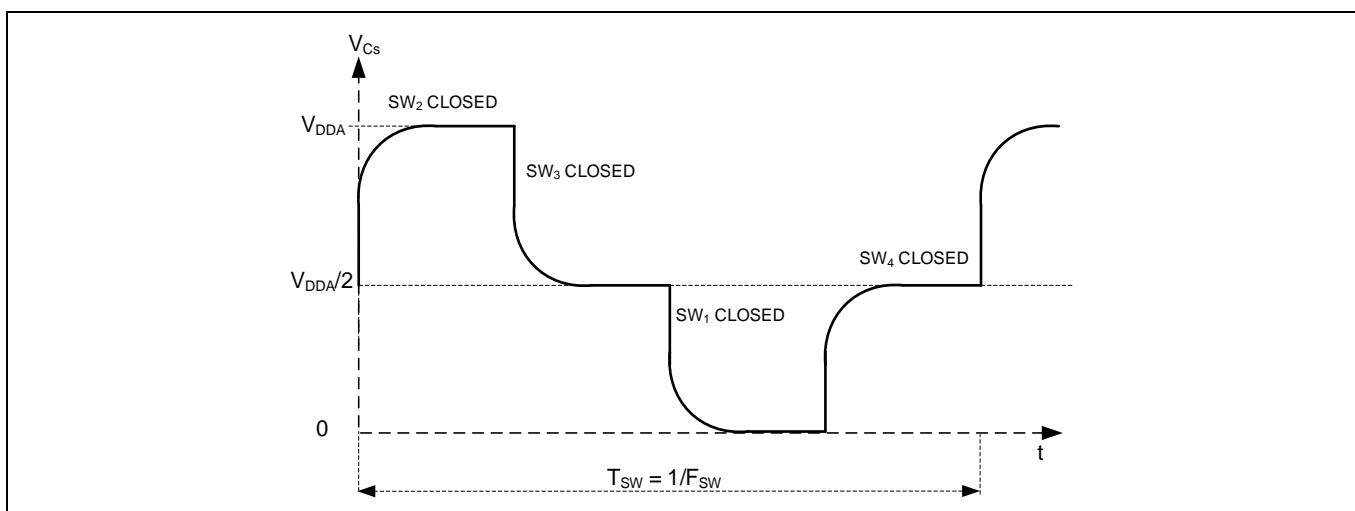


Figure 46 センサー容量の両端の電圧

3.4.2 コンデンサ DAC (CDAC)

IDAC は、第 5 世代 CAPSENSE™ アーキテクチャで CDAC に置き換えられます。これは、2 つの CDAC、リファレンスコンデンサ DAC および補償コンデンサ DAC で構成されます。各センスクロック周期で、セン

PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™

サー容量は、**GPIO セルの容量から電荷へのコンバーター**で説明されるように、 C_{MOD} 間の電圧のバランスを崩す方法で両方の C_{MOD} に電荷を転送します。両方のコンデンサ DAC は、センスクロック期間中に C_{MOD} に複数回切り替えられ、 C_{MOD} を元の電圧に戻します。基準コンデンサ DAC がバランスを取るために必要なサイクル数は、センサーの自己容量に比例します。

3.4.3 CAPSENSE™ クロックジェネレーター

このブロックは、PSoC™ デバイスファミリに応じて、高周波システムリソースクロック (HFCLK) またはペリフェラルクロック (PERI) からセンスクロック F_{SW} と変調クロック F_{MOD} を生成します。

3.4.3.1 センスクロック

CAPSENSE™ クロック生成は、セクション 3.2.4.1 で説明されているように、古い世代のクロック生成と似ています。

3.4.3.2 変調クロック

変調クロックは、**レシオメトリックセンシングテクノロジー**によって使用されます。このクロックは、**式 16** と **式 17** に基づいてセンサーのスキャン時間を決定します。

式 16. センサー スキャン時間

$$\text{センサー スキャン時間} = \text{ハードウェア スキャン時間} + \text{センサー 初期化時間}$$

式 17. ハードウェア スキャン時間

$$\text{ハードウェア スキャン時間} = \frac{\text{(サブ変換数)}}{\text{センスクロック周波数}}$$

ここで、

サブ変換数 = シングル スキャンでのサブ変換の総数

サブ変換 = センスクロック サイクル内で実行された変換をカウントする静電容量

センサー 初期化時間 = センサーが内部レジスタに書き込み、スキャンを開始するのにかかる時間

3.4.4 レシオメトリックセンシングテクノロジー

レシオメトリックコンバーターと 2 つの CDAC、リファレンスコンデンサ DAC と補償コンデンサ DAC で構成されています。各センスクロック周期で、センサー容量は、**GPIO セルの容量から電荷へのコンバーター**で説明されているように、 C_{MOD} 間の電圧のバランスを崩す方法で両方の C_{MOD} に電荷を転送します。レシオメトリックコンバーターは、2 つの C_{MOD} 間の小さな電圧変動に対応して、基準 CDAC をオンまたはオフに切り替えて、 C_{MOD} の電圧を同じレベルに維持することにより、基準 CDAC を制御します。 C_{MOD} 間の電圧のバランスを取るためにリファレンスコンデンサ DAC が必要とするサイクル数は、センサーの自己容量に比例します。

補償コンデンサは、センサーからの過剰な相互容量を補償して感度を上げるために使用されます。切り替えられる回数は、ユーザーアプリケーションがセンサーの相互容量から補償 (除去) しようとしている電荷の量によって異なります。

レシオメトリックコンバーターは、シングル CDAC モードまたはデュアル CDAC モードのいずれかで動作できます。

PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™

- シングル CDAC モードでは、基準 CDAC はレシオメトリックコンバーターによって制御されます。補償 CDAC は常にオフです。
- デュアル CDAC モードでは、基準 CDAC はレシオメトリックコンバーターによって制御されます。補償 CDAC は常にオンです。リファレンス CDAC は最大 95% を補償でき、[変換利得および CAPSENSE™ 信号](#) で説明されるように信号が増加します。

シングル CDAC モードでは、 C_{ref} が基準 CDAC の値である場合、raw カウントの概算値は [式 18](#) で与えられます。

式 18. CSD-RM シングル CDAC raw カウント

$$\text{Rawcount} = \text{Maxcount} \cdot \frac{C_s}{\text{SnsClk}_{Div} \cdot C_{ref}}$$

デュアル CDAC モードでは、補正 CDAC は常にオンになっています。 C_{comp} が補償 CDAC の場合、raw カウントの式は [式 19](#) で与えられます。

式 19. CSD-RM デュアル CDAC raw カウント

$$\text{Rawcount} = \text{Maxcount} \cdot \frac{C_s - 2 \cdot \frac{\text{SnsClk}_{Div}}{\text{CompClk}_{Div}} \cdot C_{comp}}{\text{SnsClk}_{Div} \cdot C_{ref}}$$

ここで、

$\text{MaxCount} = N_{Sub} \cdot \text{SnsClk}_{Div}$

N_{Sub} = サブ変換数

SnsClk_{Div} = センスクロック分周器

CompClk_{Div} = 補償 CDAC 分周器

C_s = センサーの静電容量

C_{ref} = 基準静電容量

C_{comp} = 補償静電容量

[式 18](#) のように、出力 raw カウントは、基準静電容量に対するセンサー静電容量の比率に比例するため、レシオメトリックセンシングという名前が付けられています。

ノイズの改善は、前世代の CAPSENSE™ テクノロジーに対する第 5 世代の主な利点の 1 つです。第 4 世代の主なノイズ源は、電流 (I_{MOD})、基準電圧 (V_{REF})、クロックジッタ (F_{SW}) です ([式 12](#) を参照)。これらのノイズソースは、第 5 世代では削除されています ([式 19](#) を参照)。IDAC は CDAC に置き換えられました。システムは完全に差動化されているため、 V_{REF} は必要ありません。スキャン結果がクロックの持続時間ではなくクロックのエッジに基づくようになったため、CAPSENSE™ アーキテクチャはジッターの影響を受けなくなりました。

3.4.5 アナログマルチプレクサ (AMUX) と制御マトリックス (CTRLMUX)

第 5 世代で導入されたもう 1 つの機能は、[Figure 44](#) に示す制御マトリックス (CTRLMUX) です。CTRLMUX は自律スキャンを可能にし、オンチップ IO ノイズに対する耐性を提供します。CTRLMUX により、CAPSENSE™ IP は (従来の GPIO モードに加えて) センサー入力⁵ を直接処理できるため、CPU なしでセンサーの自律スキャンをサポートします。

⁵ 限られた数の入力をサポートします。詳細については、[デバイスデータシート](#) を参照してください。

PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™

3.4.6 CAPSENSE™ CSD-RM シールド

PSoC™ 4 CAPSENSE™ は、液体耐性と近接検知のためのシールド電極をサポートしています。シールドの目的は、センサーとシールド電極間の寄生容量を取り除くことです。これが液体耐性にどのように役立つかについての詳細は、[被駆動シールド信号およびシールド電極および自己容量センサーに対する液滴と液体流の影響](#)を参照してください。第 5 世代 CAPSENSE™ アーキテクチャは、アクティブシールドとパッシブシールドの 2 つのシールドモードをサポートしています。

3.4.6.1 アクティブシールド

アクティブシールドモードでは、シールド回路は、[Figure 47](#) に示すように、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。

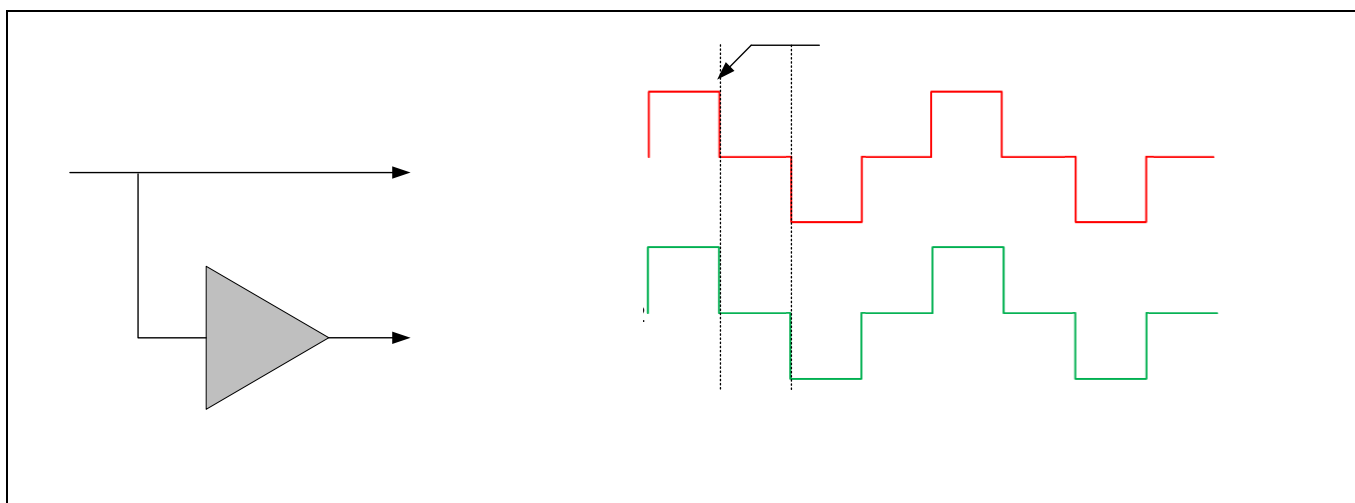


Figure 47 アクティブシールド信号

3.4.6.2 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ません。代わりに、[Figure 48](#) に示すようにシールドが VDDA 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 つのセンサクロックごとに無効になるように制御されます。

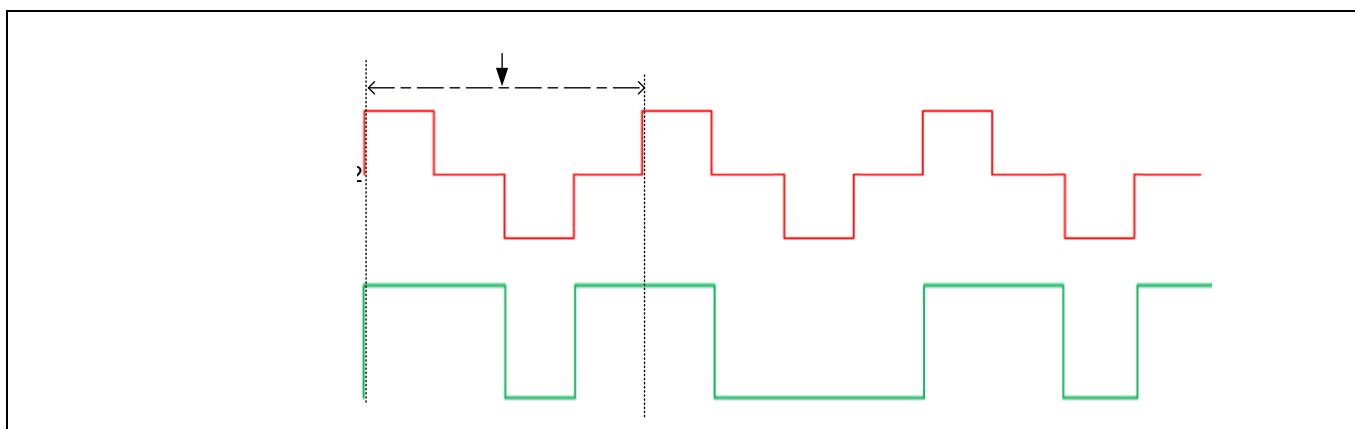


Figure 48 パッシブシールド信号

[Table 4](#) に、アクティブシールド機能とパッシブシールド機能の機能比較を示します。

PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™

Table 4 アクティブシールドとパッシブシールド

特長	アクティブシールド	パッシブシールド	影響
性能	高い	低い	高性能アプリケーションには、アクティブシールドが推奨されます。
電力影響	高い	低い	低電力アプリケーションには、パッシブシールドが推奨されます。

3.5 CAPSENSE™ CSX-RM センシング方式 (第 5 世代)

Figure 49 に、CSX-RM センシング回路を示します。実装では、次のハードウェアサブブロックを使用します。

- 2つの8ビットコンデンサ DAC とレシオメトリックコンバーター
- AMUXBUS および CTRLMUX
- Tx クロックおよび変調器クロック用の CAPSENSE™ クロックジェネレーター
- Tx および Rx 電極のポートピンおよび外部静電容量
- 2つの外部コンデンサ (C_{MOD1} および C_{MOD2})

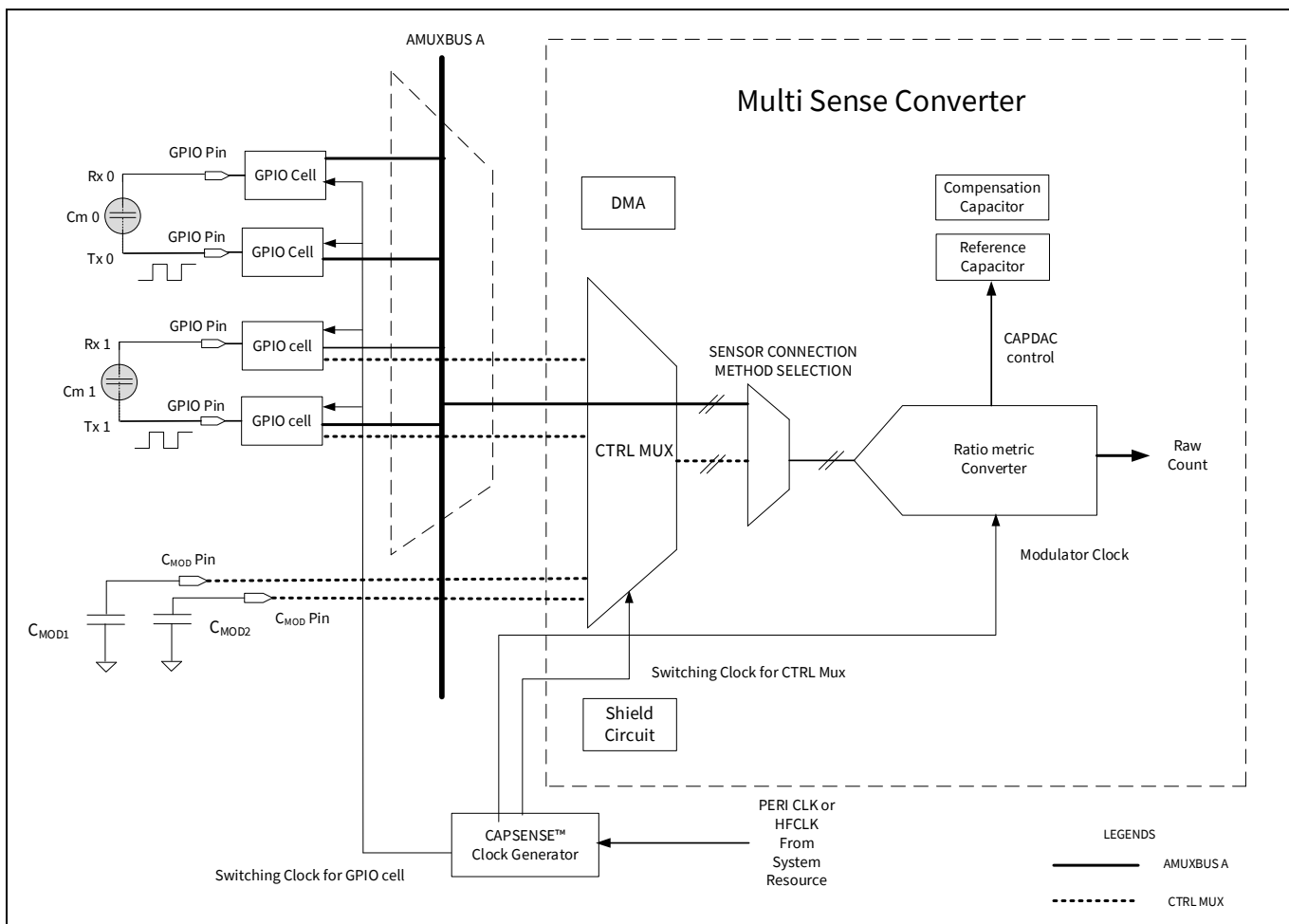


Figure 49 CAPSENSE™ CSX-RM センシング方式の設定

PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™

CSX-RM センシング方式では、**Figure 49** に示すように、Tx 電極と Rx 電極間の相互容量を測定します。Tx 電極は、 V_{DDA} とグラウンドを切り替えるデジタル波形 (Tx クロック) によってアクティブになります。Rx 電極は、AMUXBUS A または CTRLMUX に静的に接続されます。CSX-RM 方式では、 C_{MOD1} と C_{MOD2} の 2 つの外部積分コンデンサが必要です。

サンプリング (「サンプル」を生成するプロセス) は、両方の外部コンデンサ (C_{MOD}) の電圧を $V_{DDA}/2$ に初期化し、一連のサブ変換を実行することにより、ファームウェアによって開始されます。サブ変換は、Tx クロックサイクル内で実行される変換をカウントするための容量です。サンプル内のすべてのサブ変換の結果の合計は、「raw count」と呼ばれます。

Tx クロックの立ち上りエッジと立ち下りエッジで、電荷は Tx 電極から Rx 電極に流れます。それは、外部 C_{MOD} コンデンサ間の電圧のバランスを崩すような方法です。両方のコンデンサ DAC (基準および補償コンデンサ DAC) は、センスクロック期間中に C_{MOD} に複数回切り替えられ、 C_{MOD} を元の電圧に戻します。基準コンデンサ DAC がバランスを取るために必要なサイクル数は、電極間の相互容量 C_M に比例します。

基準コンデンサが変調器クロックに対して切り替えられる回数は、**式 20** に従って Tx クロック分周器の値で表されます。

式 20. Tx クロック分周器

$$TxClk_{Div} = \frac{F_{Mod}}{F_{Tx}}$$

ここで、

$TxClk_{Div}$ = Tx クロック分周器

F_{Mod} = 変調器周波数

F_{Tx} = Tx クロック周波数

補償コンデンサは、センサーからの過剰な相互容量を補償して感度を上げるために使用されます。切り替えられる回数は、ユーザーアプリケーションがセンサーの相互容量から補償 (除去) しようとしている電荷の量によって異なります。変調器クロックに対して補償コンデンサが切り替わる回数は、**式 21** による補償 CDAC 分周器の値で表されます。CDAC 補償クロック分周器は、Tx クロック分周器以下である必要があります。

式 21. 補償 CDAC 分周器

$$CompClk_{Div} = \frac{F_{MOD}}{F_{Comp}}$$

ここで、

$CompClk_{Div}$ = 補償 CDAC 分周器

F_{MOD} = 変調器周波数

F_{comp} = 補償 CDAC クロック周波数

3.5.1 レシオメトリックセンシングテクノロジー

レシオメトリックコンバーターは、各スキャン後のセンサーの相互容量に比例する同等の raw カウントを提供します。レシオメトリックコンバーターは、シングル CDAC モードまたはデュアル CDAC モードのいずれかで動作できます。

PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™

- シングル CDAC モードでは、基準 CDAC はレシオメトリックコンバーターによって制御されます。補償 CDAC は常にオフです。
- デュアル CDAC モードでは、基準 CDAC はレシオメトリックコンバーターによって制御されます。補償 CDAC は常にオンです。補償 CDAC は最大 95% を補償でき、[変換利得および CAPSENSE™ 信号](#)で説明されるように信号が増加します。

シングル CDAC モードでは、 C_{ref} が基準 CDAC の値である場合、raw カウントの概算値は [式 22](#) で与えられます。

式 22. CSX-RM シングル CDAC raw カウント

$$\text{Rawcount} = \text{Maxcount} \cdot \frac{C_M}{\text{TxClk}_{Div} \cdot \left(\frac{C_{ref}}{2}\right)}$$

デュアル CDAC モードでは、補償 CDAC は常にオンになっています。 C_{comp} が補償 CDAC の場合、raw カウントの式は [式 23](#) で与えられます。

式 23. CSX-RM デュアル CDAC raw カウント

$$\text{Rawcount} = \text{Maxcount} \cdot \frac{C_M - \frac{\text{TxClk}_{Div}}{\text{CompClk}_{Div}} \cdot C_{comp}}{\text{TxClk}_{Div} \cdot \left(\frac{C_{ref}}{2}\right)}$$

ここで、

$\text{MaxCount} = N_{Sub} \cdot \text{TxClk}_{Div}$

N_{Sub} = サブ変換数

TxClk_{Div} = Tx クロック分周器

CompClk_{Div} = CDAC 補償分周器

C_M = センサーの相互容量

C_{ref} = 基準容量

C_{comp} = 補償容量

[式 23](#) によると、出力 raw カウントは、基準容量に対するセンサーの相互容量の比率に比例するため、レシオメトリックセンシングと呼ばれます。

3.6 自律スキャン

前世代の CAPSENSE™ テクノロジーでは、各スキャンの後、CPU は次のセンサーを構成するために中断されます。第 5 世代 CAPSENSE™ テクノロジーの自律スキャンモードは、次のすべてのセンサーをスキャンするための CPU の介入を回避します。これにより、多数のセンサーを備えたウィジェットのスキャンに必要な CPU 帯域幅が大幅に削減されます。自律スキャンには、CTRLMUX や DMA などの機能が必要です。CTRLMUX でサポートされるピンの数が制限されているため、自律スキャンをサポートするピンの数も制限されます。詳細については、[自律スキャンの設定](#) セクションを参照してください。

3.7 マルチチャネルの使用

PSoC™ 4100S Max デバイスは、MSC0 と MSC1 の 2 つの第 5 世代 CAPSENSE™ ブロックをサポートします。各ブロックは、[CAPSENSE™ CSD-RM センシング方式 \(第 5 世代\)](#) および [CAPSENSE™ CSX-RM センシング方式 \(第 5 世代\)](#) のセクションで説明したものと同一機能とパフォーマンスを備えます。各インスタ

PSoC™ 4 および PSoC™ 6 MCU CAPSENSE™

ンスはチャンネルと見なすことができ、複数のインスタンスは複数のチャンネルを意味します。マルチチャンネルの動作は、シングルチップ内の複数のインスタンスまたは複数のチップを持つことでサポートできます。チャンネルの動作は同期され、チャンネルに接続されたセンサーをスキャンするときにロックステップで動作します。Lockstep は、クロック同期を保証し、同期されていないセンスクロックによるクロスチャンネルノイズを回避します。

詳細については、[マルチチャンネルスキャンニング](#)のセクションを参照してください。

CAPSENSE™ 設計および開発ツール

4 CAPSENSE™ 設計および開発ツール

ここでは、CAPSENSE™ アプリケーションを開発するために、PSoC™ Creator や ModusToolbox™ などの利用可能なソフトウェアツールを紹介합니다。詳細については、各 IDE のユーザーマニュアルを参照してください。Table 5 に、PSoC™ Creator および ModusToolbox™ でサポートされるデバイスおよび CAPSENSE™ コンポーネント/ミドルウェアのバージョンを示します。

Table 5 ツールと対応デバイス

デバイス	ソフトウェアツール	CAPSENSE™ ライブラリ
PSoC™ 4000S, PSoC™ 4100S, PSoC™ 4100S Plus, PSoC™ 4100S Plus 256K, PSoC™ 4500S	ModusToolbox™, PSoC™ Creator	CAPSENSE™ middleware, CAPSENSE™ component
PSoC™ 4100S Max, すべての PSoC™ 6 デバイス	ModusToolbox™	CAPSENSE™ middleware
他のすべての PSoC™ 4 デバイス	PSoC™ Creator	CAPSENSE™ component

4.1 PSoC™ Creator

PSoC™ Creator は最先端で使いやすい IDE です。これは、従来の回路図入力に基づいて、ハードウェア構成とソフトウェア開発のユニークな組み合わせを提供します。コンポーネントのライブラリを使用してドラッグ&ドロップの設計環境でアプリケーションが開発できます。詳細は [PSoC™ Creator ホームページ](#) を参照してください。

4.1.1 CAPSENSE™ コンポーネント

PSoC™ Creator は CAPSENSE™ コンポーネントを提供します。これは単にコンポーネントを設定するだけで PSoC™ の静電容量タッチシステムを作成できます。CAPSENSE™ コンポーネントは、ファームウェア開発を簡素化するために、アプリケーションプログラミング インターフェース (API) も提供します。いくつかの PSoC™ 4 Bluetooth® LE および PSoC™ 6 MCU デバイスは CAPSENSE™ ジェスチャーコンポーネントもサポートします (デバイスがこのコンポーネントをサポートするかを確認するためには、該当する [デバイス データシート](#) を参照してください)。

CAPSENSE™ 設計および開発ツール

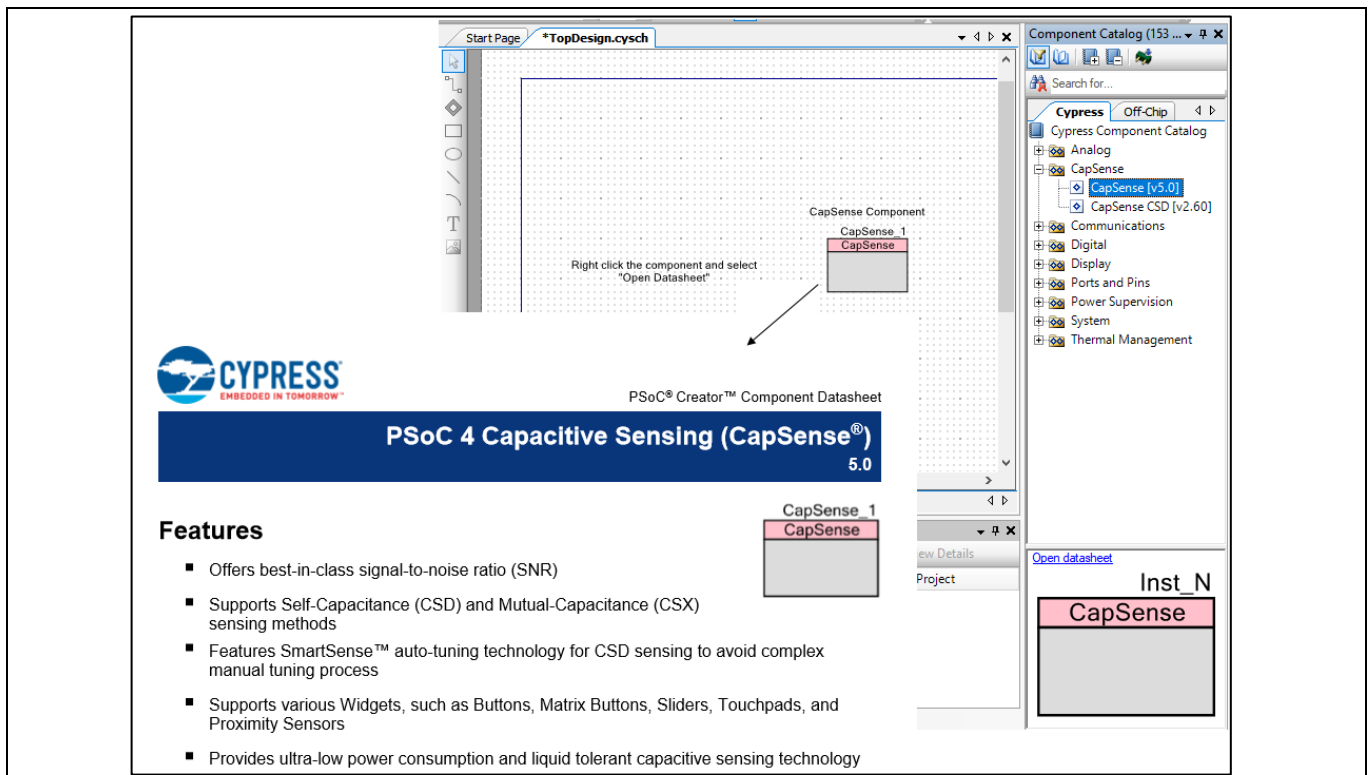


Figure 50 PSoC™ Creator コンポーネントの配置

各 CAPSENSE™ コンポーネントには、コンポーネントを詳細に説明する関連データシートがあります。コンポーネントを右クリックして **Open Datasheet** を選択すると、コンポーネントのデータシートを開きます。

CAPSENSE™ コンポーネントはチューニングプロセスを支援するために、**チューナーGUI** と呼ばれるチューナーGUI を備えます。

4.1.2 CapSense_ADC⁶ コンポーネント

CapSense_ADC コンポーネントは PSoC™ 4S シリーズ、PSoC™ 4100S Plus、PSoC™ 4100PS、および PSoC™ 6 MCU デバイスにのみ適用されます。CAPSENSE™ および ADC 動作の両方を必要とされる場合、このコンポーネントを使用することが必要です。このコンポーネントを使用すると、CAPSENSE™ ブロックを ADC の動作とタッチ機能に時分割多重化で使用できます。

4.1.3 チューナーGUI

チューナーヘルパーは **CAPSENSE™** コンポーネントに含まれ、CAPSENSE™ パラメーターのチューニング、および raw カウント、ベースライン、および差分カウントなどのセンサーデータの監視に役立ちます。チューナーGUI の使用手順の詳細は該当 **コンポーネントデータシート/ミドルウェア文書** を参照してください。

4.1.4 サンプルプロジェクト

回路図入力およびファームウェア開発を学ぶために、PSoC™ Creator で提供される CAPSENSE™ のサンプルプロジェクトを使用できます。CAPSENSE™ サンプルプロジェクトを検索するために、**Figure 51** に示

⁶ CapSense_ADC は、第 5 世代 CAPSENSE™ ブロックを備えたデバイスではサポートされません。

CAPSENSE™ 設計および開発ツール

すように、PSoC™ Creator のスタート ページに移動し、**Find Code Example...** をクリックして、適切なアーキテクチャを選択します。プロジェクト名の一部または全部を **Filter by** 領域に書いて、プロジェクトのフィルタリングもできます。

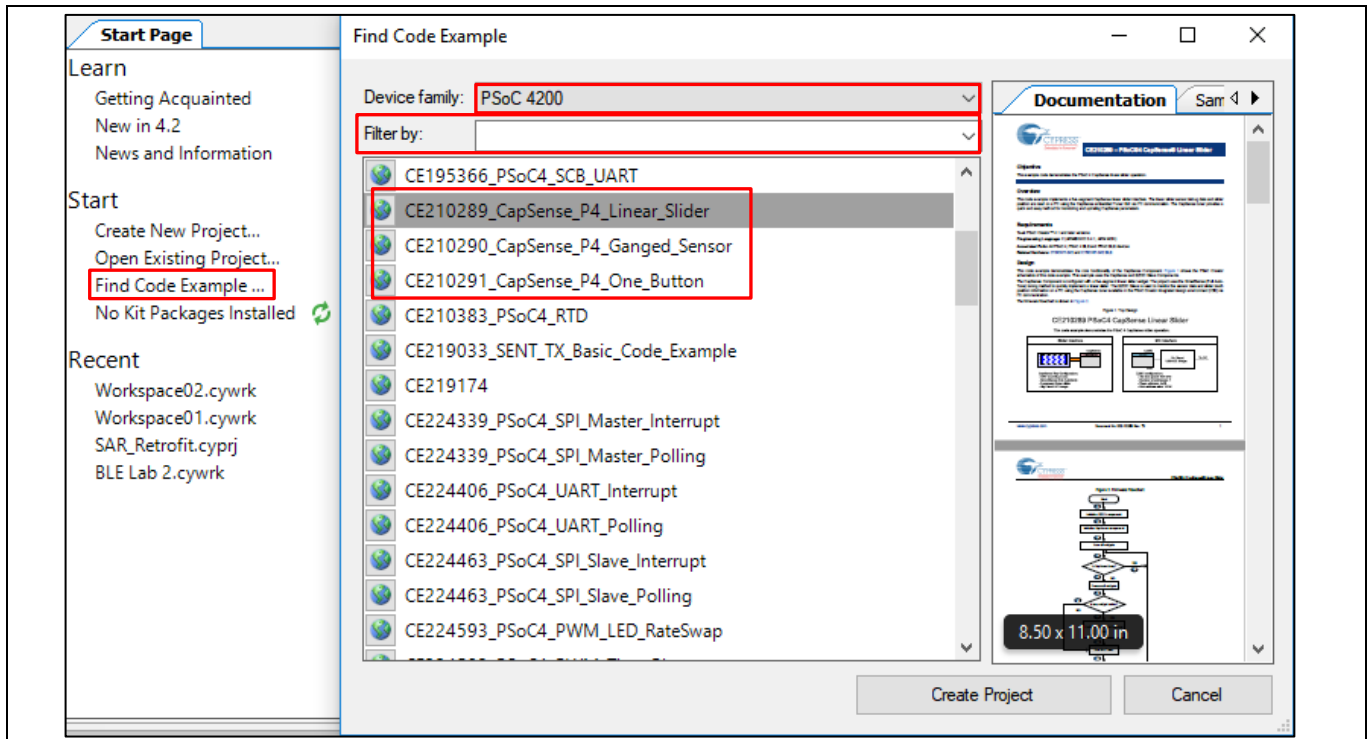


Figure 51 PSoC™ Creator のサンプル プロジェクト

4.2 ModusToolbox™

PSoC™ 6 および PSoC™ 4⁷ ベースの CAPSENSE™ アプリケーションを開発するために ModusToolbox™ ソフトウェア スイートが使われます。こちらから ModusToolbox™ をダウンロードできます。このソフトウェアを使用する前に、[クイック スタート ガイド](#) および [ユーザー ガイド](#) を読まれることを推奨します。ModusToolbox™ IDE をシステムにインストールすることで、ModusToolbox™ に対応するデバイスの CAPSENSE™ アプリケーションを作成できます。

4.2.1 CAPSENSE™ ミドルウェア

ModusToolbox™ は、CAPSENSE™ ミドルウェアを提供します。これは、CAPSENSE™ 構成ツールでパラメータを構成するだけで、PSoC™ で静電容量式タッチシステムを作成するために使用できます。ミドルウェアは、ファームウェア開発を簡素化するためのアプリケーションプログラミングインターフェース (API) も提供します。詳細については、[CAPSENSE™ middleware library](#) を参照してください。

4.2.2 CAPSENSE™ 設定ツール

[ModusToolbox™](#) の CAPSENSE™ 設定ツールは、CAPSENSE™ ハードウェアおよびソフトウェアパラメータを構成するために使用される [PSoC™ Creator](#) のツールに似ています。ModusToolbox™ での CAPSENSE™ の構成の詳細については、[ModusToolbox™ CAPSENSE™ configurator guide](#) および [CAPSENSE™ middleware library](#) を参照してください。Figure 52 に、ModusToolbox™ で CAPSENSE™ 構成ツールを開く方法を示します。または、ModusToolbox™ の **Quick** パネルから開くこともできます。ドキュメントを

⁷ ModusToolbox™ でサポートされている PSoC™ 4 デバイスについては、[Table 5](#) を参照してください。

CAPSENSE™ 設計および開発ツール

簡単にするために、このデザインガイドでは、PSoC™ Creator CAPSENSE™ コンポーネントで CAPSENSE™ パラメーターを選択する方法を示します。

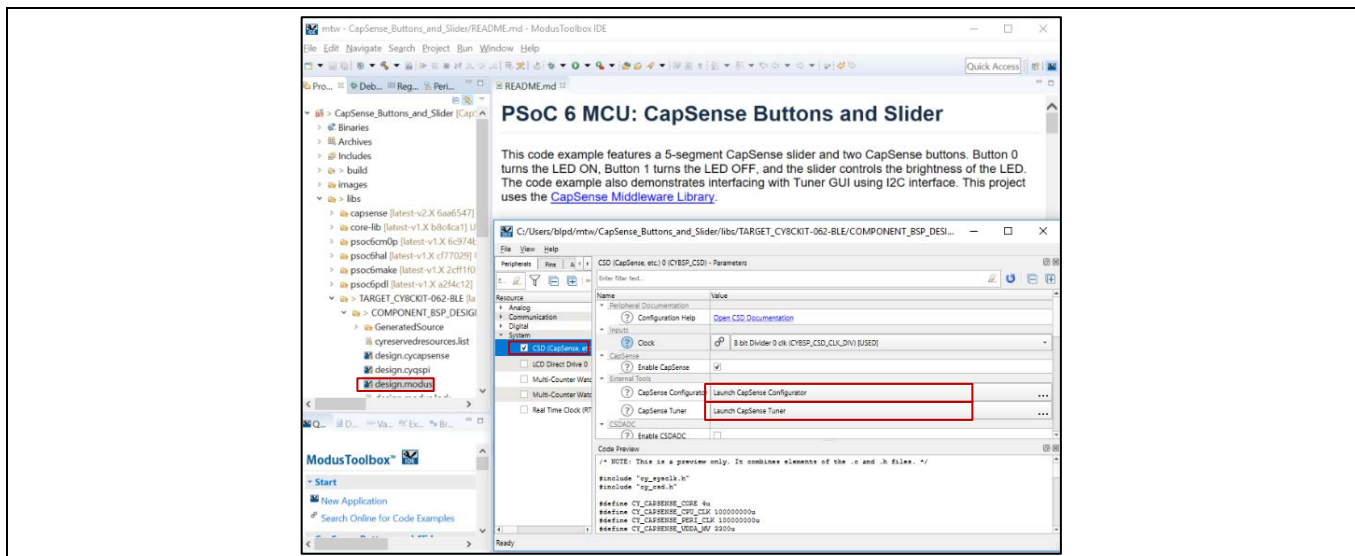


Figure 52 ModusToolbox™での CAPSENSE™ 構成ツール

4.2.3 CSDADC ミドルウェア⁸

この CSDADC ミドルウェアは、CAPSENSE™ と ADC の両方の操作が必要な場合に使用する必要があります。このミドルウェアを使用すると、CAPSENSE™ ハードウェアブロックを ADC の動作とタッチ機能に時分割多重化で使用できます。これは、CSD、ADC、CSX の 3 つのセンシングモードすべてに使用できます。詳細については、[CSDADC middleware library](#) のドキュメントを参照してください。

4.2.4 CSDIDAC ミドルウェア

CSDIDAC ミドルウェアを使用すると、CAPSENSE™ IDAC をスタンドアロンモードで使用できます。CAPSENSE™ ミドルウェアを使用しない場合、または CAPSENSE™ に IDAC を 1 つだけ使用する場合は、このミドルウェアを使用できます。[CSDIDAC middleware library](#) のドキュメントを参照してください。

4.2.5 CAPSENSE™ チューナー

ModusToolbox™ は、CAPSENSE™ パラメーターの調整に使用できる GUI ツールにも対応します。このツールは、[Figure 52](#) に示すように、*Launch CapSense Tuner* を選択することで [Device configurator](#) から開けます。[CAPSENSE™™ tuner guide](#) のドキュメントを参照してください。

4.2.6 サンプルプロジェクト

CAPSENSE™ システム設計をすばやく開始するためには、ModusToolbox™ で提供されているサンプルプロジェクトから始めてください。**File > New > ModusToolbox Application** に移動してください。CAPSENSE™ サンプルプロジェクトがあります。デバイスに適切なボードサポートパッケージを選択してください。[Figure 53](#) に、使用可能なサンプルコードのリストから ModusToolbox™ で CAPSENSE™ CSD ボタンサンプル開始コードを作成する方法を示します。

⁸ CapSense_ADC は、第 5 世代 CAPSENSE™ ブロックを備えたデバイスではサポートされません。

CAPSENSE™ 設計および開発ツール

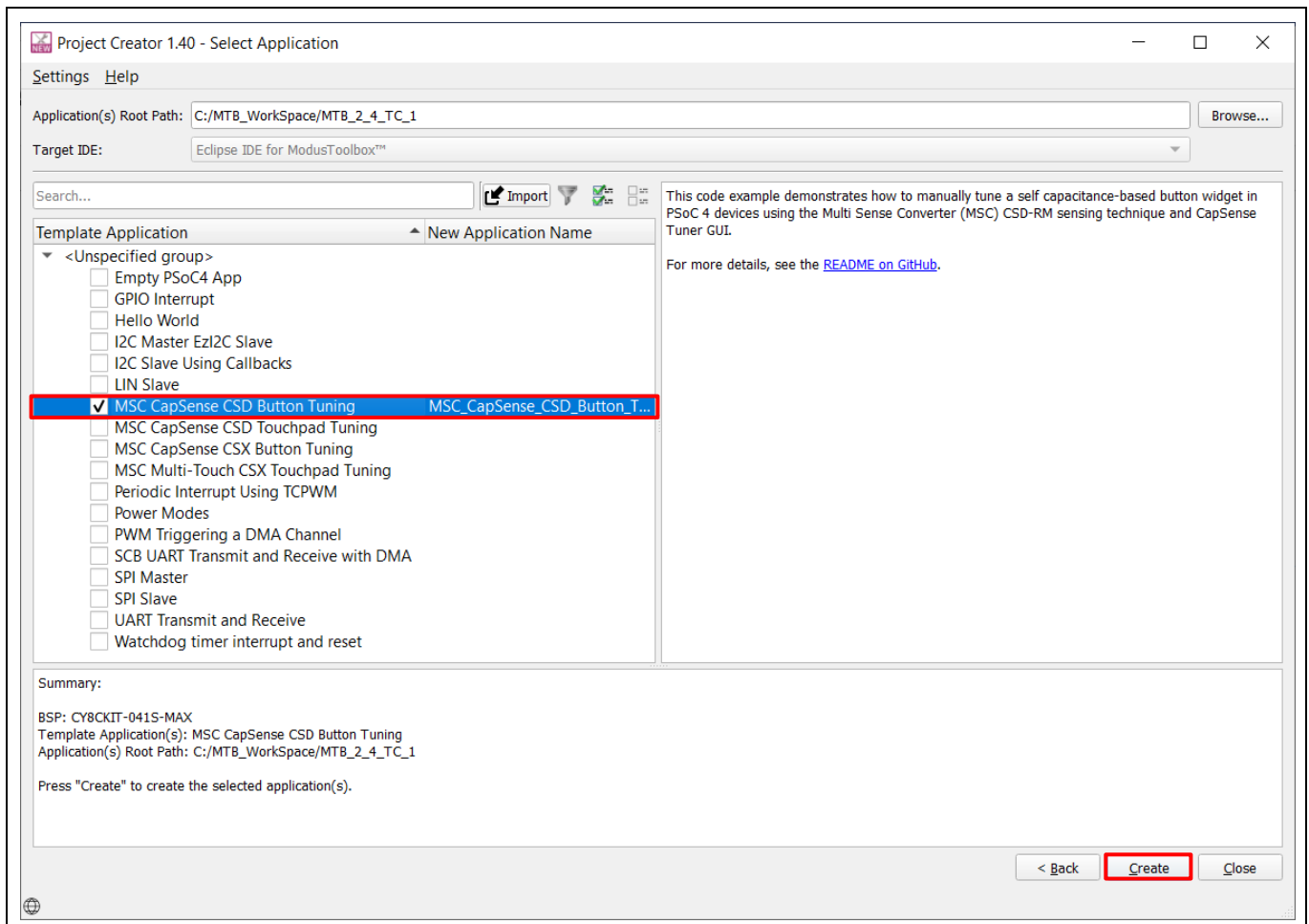


Figure 53 ModusToolbox™での CAPSENSE™ CSD ボタンサンプルプロジェクトの作成

4.3 ハードウェアキット

Table 6 に PSoC™ 4 および PSoC™ 6 CAPSENSE™ の評価をサポートする開発キットの一覧を示します。

Table 6 PSoC™ 4 および PSoC™ 6 CAPSENSE™ の開発キット

開発キット	対応する CAPSENSE™機能
PSoC™ 4000 Pioneer Kit (CY8CKIT-040)	1 個の 5x6 CAPSENSE™ タッチパッドおよび 1 個のワイヤ近接センサー
PSoC™ 4 S シリーズ Pioneer Kit (CY8CKIT-041)	2 個の自己容量または相互容量センシング ボタン 1 個の 7x7 自己容量または相互容量センシング タッチパッド
PSoC™ 4 S シリーズ プロトタイピング キット (CY8CKIT-145)	3 個の自己容量または相互容量センシング ボタン 1 個の 5 素子自己容量または相互容量センシング リニア スライダー
PSoC™ 4100S Plus プロトタイピング キット (CY8CKIT-149)	3 個の自己容量または相互容量センシング ボタン 1 個の 6 素子自己容量または相互容量センシング リニア スライダー
PSoC™ 4100S Max pioneer kit (CY8CKIT-041S-Max)	2 個の自己容量または相互容量センシング ボタン

CAPSENSE™ 設計および開発ツール

開発キット	対応する CAPSENSE™ 機能
	1 個の 8 素子の自己容量または相互容量センシング グリニアスライダ 1 個の 10x16 自己容量または相互容量センシング タッチパッド 1 個の近接センサー ループ
PSoC™ 4 Pioneer Kit (CY8CKIT-042)	1 個の 5 素子リニア スライダ
PSoC™ 4 Bluetooth® LE Pioneer Kit (CY8CKIT-042-BLE)	1 個の 5 素子リニア スライダ および 1 個のワイヤ近接センサー
PSoC™ 4200-M Pioneer Kit (CY8CKIT-044)	1 個の 5 素子ジェスチャー検出 および 2 個のワイヤ近接センサー
PSoC™ 4200-L Pioneer Kit (CY8CKIT-046)	1 個の 5 素子ジェスチャー検出、2 個のワイヤ近接センサー および 1 個の 8 素子ラジアル スライダ
PSoC™ 4100PS プロトタイピング キット (CY8CKIT-147)	基板搭載の CAPSENSE™ センサーなし。このキットは外部センサーを任意の I/O ピンに接続するために使用できます。
CAPSENSE™ 近接シールド (CY8CKIT-024)	1 個の 4 素子ジェスチャー検出 および 1 個のループ近接センサー
CAPSENSE™ 液体レベルセンシングシールド (CY8CKIT-022)	1 個の 2 素子フレキシブル PCB および 12 素子フレキシブル PCB
PSoC™ 4 プロセッサ モジュール (CY8CKIT-038)、PSoC™ 開発キット (CY8CKIT-001)	1 個の 5 素子リニア スライダ および 2 個のボタン
CY8CKIT-038 および CY8CKIT-001 に使用する CAPSENSE™ 拡張基板キット (CY8CKIT-031)	1 個の 10 素子スライダ、5 個のボタン、および LED 表示付き 4x4 マトリックス ボタン
MiniProg3 プログラムおよびデバッグキット (CY8CKIT-002)	CY8CKIT-038 の CAPSENSE™ 性能チューニング
PSoC™ 6 Wi-Fi BT Pioneer Kit (CY8CKIT-062-WiFi-BT pioneer kit) および PSoC™ 6 Bluetooth® LE Pioneer Kit (CY8CKIT-062-BLE pioneer Kit)	1 個の 5 素子 CAPSENSE™ スライダ、2 個の CAPSENSE™ ボタン、1 個の CAPSENSE™ 近接センシング ヘッド、1 個の近接センサー
PSoC™ 6 Wi-Fi BT プロトタイピング キット (CY8CPROTO-063-4343W)	1 個の 5 素子 CAPSENSE™ スライダ および 2 個の相互容量 CAPSENSE™ ボタン

5 CAPSENSE™性能のチューニング

センサーのレイアウトが終わると ([PCB レイアウト ガイドライン](#)を参照してください)、次のステップではファームウェアを実装し、最適な性能を得るためにセンサーの CAPSENSE™パラメーターをチューニングします。CAPSENSE™センシング方式はハードウェアとファームウェア技術を結合したものです。したがって、適切な動作に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必要です。これらのパラメーターは、信頼できるタッチ検出と高速応答に対応した最適値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。SmartSense (自動チューニングとしても知られる) と呼ばれるユニークな機能が PSoC™ 4 および PSoC™ 6 CAPSENSE™で使用可能です。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。

5.1 SmartSense および手動チューニングの選択

SmartSense 自動チューニングは開発期間を短縮し、PCB のばらつきにわたって安定した性能を提供します。しかし、[コンポーネント データシート/ミドルウェア文書](#)または [ModusToolbox™ CAPSENSE™ configurator guide](#) に示すように、実行時に CAPSENSE™パラメーターのチューニングを行うために、追加の RAM と CPU リソースを必要とします。SmartSense はシンプル ボタンおよびスライダー ウィジェットに関わる従来の CAPSENSE™アプリケーションに主に推奨されますが、現在、[自己容量センシング](#)のみに対応し、[相互容量センシング](#)に対応していません。

それに反して、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作業が必要ですが、それにより応答時間と消費電力など静電容量センシング システムの特性を厳密に制御できます。これは従来のボタンおよび近接、液体レベルセンシングなどのスライダー アプリケーションを超えて、CAPSENSE™の使用を可能にします。

SmartSense は、従来のすべての CAPSENSE™アプリケーションについて推奨のチューニング方法です。設計が次の要件を満たしている場合は、SmartSense 自動調整を使用する必要があります。

- デザインが、ボタン、スライダー、タッチパッドなどの従来のユーザーインターフェースアプリケーション用。
- センサーの寄生容量 (C_p) が、[コンポーネント データシート/ミドルウェア文書](#)または [ModusToolbox™ CAPSENSE™ configurator guide](#) の「SmartSense 動作条件」セクションに記載されている SmartSense サポート範囲内。
- SmartSense によって選択されたセンサースキャン時間が、エンドシステムの応答時間/電力要件を満たす。
- SmartSense 自動チューニングが、デザインの RAM/flash 要件を満たす。

他のすべてのアプリケーションでは、[手動チューニング](#)を使用してください。このような場合、SmartSense を最初のステップとして使用して、センスクロック周波数などの最適なハードウェアパラメーターを見つけてから、チューニングモードを手動チューニングに変更して CAPSENSE™パラメーターをさらにチューニングすることもできます。[SmartSense を使用したハードウェアパラメーターの決定](#)を参照してください。手動チューニングは、I²C または UART とホスト PC との通信を必要とすることに注意してください。

5.2 SmartSense

5.2.1 概要

CAPSENSE™アルゴリズムは、PSoC™内のハードウェアブロックとファームウェアブロックの組合せです。したがって、適切な動作に必要ないくつかのハードウェアおよびファームウェアパラメーターがあ

CAPSENSE™性能のチューニング

ります。これらのパラメーターは、信頼性の高いタッチ検出と高速応答のために最適な値に調整する必要があります。

SmartSense は、ユーザー指定の指容量値に基づいて、最適なパフォーマンスのためのセンシングパラメーターを自動的に設定し、システム、製造、および環境の変化を継続的に補正する CAPSENSE™チューニング方法です。

SmartSense は現在、**CSD (Self-cap)** センシングモードのウィジェットのみに対応することに注意してください。**CSX (Mutual-cap)** ウィジェットは手動で調整する必要があります。

手動チューニングとは対照的に、次のように SmartSense にはいくつかの利点があります。

- **設計サイクルタイムの短縮:** 静電容量タッチのアプリケーションに応じた設計フローでは、すべてのセンサーのチューニングを含んでいます。設計で多くのセンサーが存在する場合、このステップは時間がかかることがあります。その上、設計、PCB レイアウト、メカニカル設計に変更があるときには、チューニングを繰り返す必要があります。自動チューニングは、すべてのパラメーターを自動的に設定することで、それらの問題を解決します。Figure 54 に SmartSense を使用する場合と使用しない場合の一般的な CAPSENSE™アプリケーションの設計フローを示します。

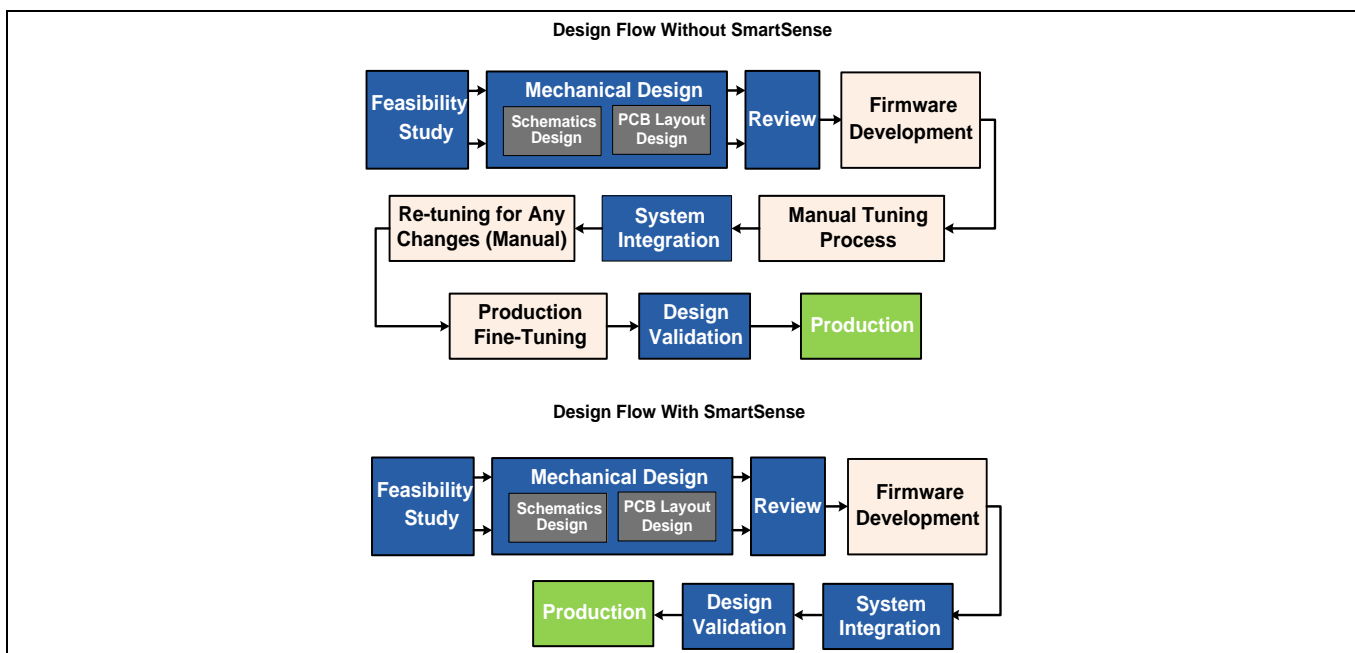


Figure 54 SmartSense を使用する場合と使用しない場合の設計フロー

- **性能は PCB の変動に依存しない:** 個々のセンサーの寄生容量 C_p は、PCB 製造工程のばらつき、または複数の業者からの供給網でベンダー間のばらつきによって異なります。製品のバッチ間で C_p に大幅な違いがある場合は、CAPSENSE™のパラメーターは、製造バッチごとに再調整しなければなりません。SmartSense は各デバイス用にパラメーターを自動的に設定するため、 C_p のばらつきの影響を低減できます。
- **使い易さ:** SmartSense では、CAPSENSE™の基本的な知識のみが必要なため、より速く、使い易くなります。

SmartSense はさまざまな方法で使用できることに注意してください。

1. **SmartSense (全自動チューニング)** - これはチューニングするための最速な方法です。この方法は実行時に、CAPSENSE™ハードウェアおよびソフトウェアパラメーターを自動的に補正します。これはほとんどの設計で推奨される手法です。

CAPSENSE™性能のチューニング

2. **SmartSense (ハードウェアパラメーターのみ)** - この方法は CAPSENSE™のすべてのハードウェアパラメーターを自動チューニングしますが、ユーザー定義のしきい値 (**Table 11** を参照) を設定できません。この方法はフラッシュ/RAM リソースの消費が SmartSense (フル自動チューニング) より少なくなります。また、この方法は自動しきい値の計算に必要な追加処理を回避します。したがって、特定スキャン速度の消費電力が少なくなります。低消費電力、ノイズのある設計、メモリ容量に制約がある要件に対して、この方法を使用します。
3. **初期チューニング用の SmartSense** - CAPSENSE™基板に最適な設定をすばやく調べて、それから手動チューニングに変えるために、初期チューニングとして SmartSense を使用することもできます。この方法は応答時間または電力消費量に対してより厳しい要件がある場合に役立ちます。最初から手動チューニングを開始する代わりに、これは最適な設定を調べるための素早い方法です。詳細については、**SmartSense を使用したハードウェアパラメーターの決定**のセクションを参照してください。

パラメーター	全自動チューニングモード	ハードウェアパラメーターのみのモード
スキャン分解能	CAPSENSE™初期化時に 1 回計算されます	
補正 IDAC		
変調 IDAC		
センスクロック周波数		
変調クロック周波数		
指しきい値	選択された指の静電容量に基づいて、CAPSENSE™初期化時に 1 回計算され、各センサースキャン後に更新されます。	手動選択。 Table 11 を参照してください。
ノイズしきい値		
ヒステリシス		
負のノイズしきい値		
低ベースラインリセット		

5.2.2 SmartSense の全自動チューニング

SmartSense 全自動チューニングモードでは、ユーザーが調整する必要のあるパラメーターは、指の静電容量パラメーターのみです。Finger Capacitance パラメーター (C_F) は、CAPSENSE™コンポーネントによって有効なタッチとして検出される必要がある指容量の最小値を示します。指がボタンセンサーに触れたときに追加される実際の C_F が、コンポーネント構成ウィンドウの Finger Capacitance パラメーターに指定された値より大きい場合は常に、センサーのステータスが '1' に変わります。ただし、指で触れて追加された実際の C_F が、コンポーネント構成ウィンドウで指定された値よりも小さい場合、センサーのステータスは '0' のままです。指の静電容量を調整する方法は、ボタンウィジェットとスライダウィジェットで異なります。

SmartSense 自動チューニングの場合でも、CAPSENSE™コンポーネントを使用すると、補正 IDAC の有効化/無効化、フィルター、耐液性関連パラメーターなどのシールド、変調器クロックなどの一般的なパラメーターを手動で構成できます。これらは、ほとんどの場合、デフォルト値のままにすることも、このガイドのそれぞれのセクションに基づいて構成することもできます。

5.2.2.1 チューニングボタンウィジェット

ここでは、ボタンウィジェットの指容量値を選択する方法について説明します。作業中の設計では、指の静電容量パラメーターの大まかな調整のみを実行することも、あるいは指の静電容量値を微調整することもできます。大まかな調整はほとんどの設計の要件を満たしますが、微調整によって、SmartSense を使用した最も効率的な CAPSENSE™パラメーター (つまり、最小センサースキャン時間) を設定できます。

CAPSENSE™性能のチューニング

C_F の値がわからない場合 (C_F は [式 1](#) に基づいて推定できます)、フィンガー容量を次のように設定してください。

1. (リストで使用可能なオプションから) 指の静電容量の最大値を指定することから始め、ボタンがタッチされたときの SNR とボタンのステータスを確認してください。 [チューナー GUI](#) を使用して SNR を見つけてください。
2. タッチ時にボタンのステータスが '1' に変わり、SNR が 5 を超えるまで、指の静電容量パラメータ値を減らしてください。 [Figure 55](#) に、設計で指の静電容量パラメータの正しい値を見つけるための詳細な手順を示します。

設定された指の静電容量がすでにコンポーネントでサポートされている最小の指の静電容量にあるときに、1 つ以上のセンサーの SNR が 5 : 1 未満の場合は、フィルターを有効にしてください。外部から誘発されたノイズが SNR の低下を引き起こしている場合は、フィルターを有効にすることもできます。この場合、適切なフィルターを選択するためには、[Table 7](#) を参照してください。CAPSENSE™コンポーネントでは、メディアンフィルター、IIR フィルター、平均フィルターなど、さまざまなタイプのフィルターを使用できます。要件に応じて、複数のフィルターを有効にして、raw カウントのノイズを減らせます。

IIR フィルターを使用する場合は、フィルター係数の値が高いフィルターを選択することから始め、5 : 1 以上の SNR が得られるまでフィルター係数を減らし続けてください。フィルターを使用すると、応答時間に影響します。応答時間と SNR 要件が満たされるように、フィルター係数を適切に選択する必要があります。

指の静電容量の最小許容値と適切なフィルターを選択しても SNR が 5:1 未満の場合は、より詳細な問題のデバッグについて [PCB レイアウト](#), [手動チューニング](#), または [チューニングデバッグ FAQ](#) を参照してください。

CAPSENSE™性能のチューニング

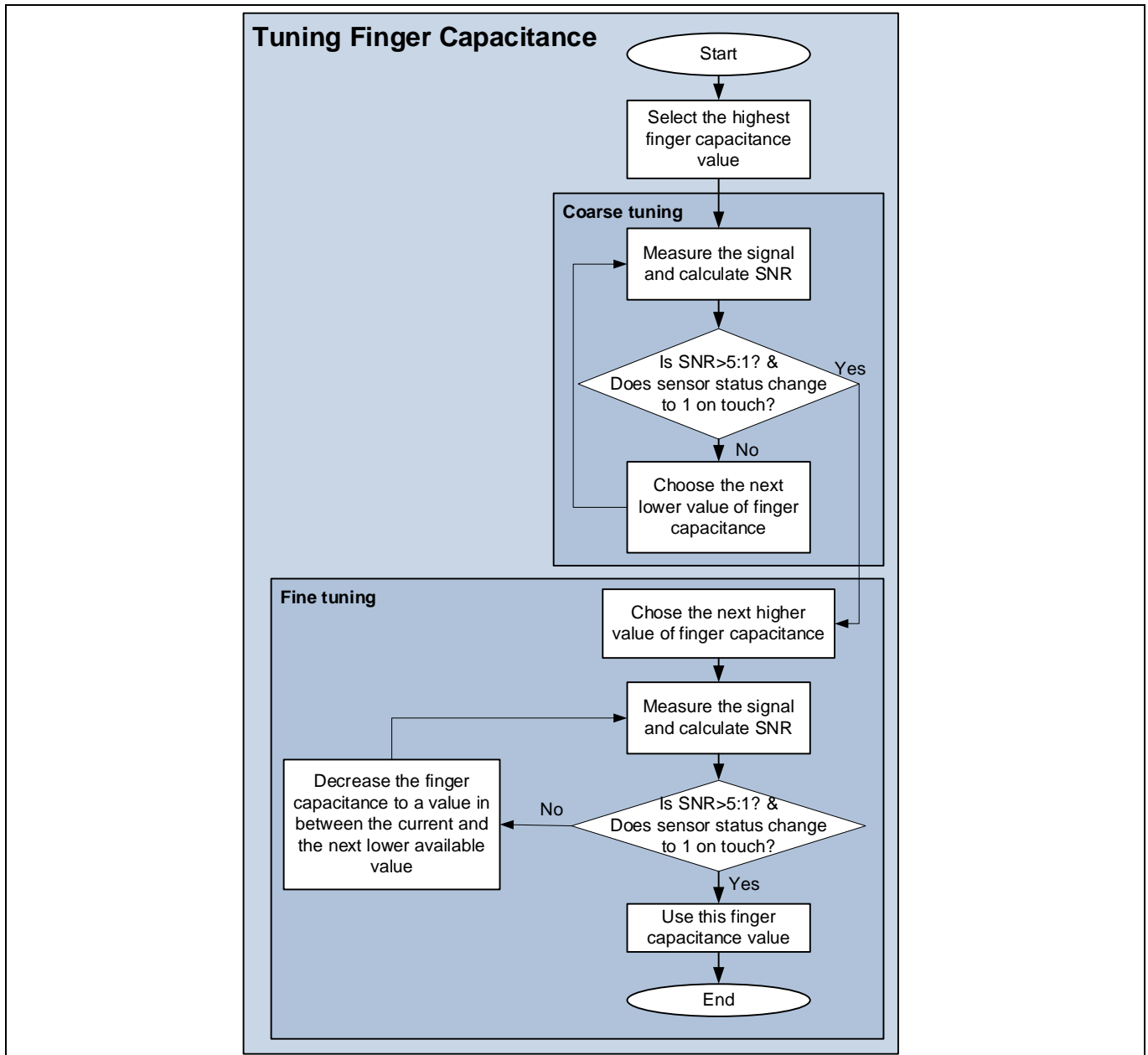


Figure 55 PSoC™ Creator での SmartSense 自動調整ベースの CAPSENSE™プロジェクトの使用

CAPSENSE™性能のチューニング

Table 7 CAPSENSE™コンポーネントの raw データ ノイズ フィルター

フィルター	説明	数学的記述	応用
中央値	3つの最新サンプルを取得して、中央値を算出する非線形フィルター	$y[i] = \text{median}(x[i], x[i-1], x[i-2])$	モーターおよびスイッチング電源からのスパイクノイズを除去
平均	等しく加重された係数を持つ、有限インパルス応答フィルター(フィードバックなし)。これは4つの最新サンプルを取得して、その平均値を算出します。	$y[i] = \frac{1}{4} * (x[i] + x[i-1] + x[i-2] + x[i-3])$	周期性ノイズ除去(例えば、電源からのノイズ)
1次 IIR	RC ローパス フィルターに類似したステップ応答を備えた無限インパルス応答フィルター(フィードバックあり)。これによって、低周波信号(指タッチ応答)を通過させます。 K 値は 256 に固定されます。 N は IIR フィルター raw カウントの係数です。低い N 値はより小さいノイズになりますが、応答速度は低下します。	$y[i] = \frac{1}{K} * \{N * x[i] + (K - N) * y[i-1]\}$	高周波ノイズ除去

5.2.2.2 チューニング スライダー ウィジェット

スライダーの場合、最初に指の静電容量を最大値に設定してください。スライダー上で指をスライドさせてください。スライダーの任意の位置で、少なくとも1つのスライダーセグメントステータスが ON で、SNR が 5:1 を超え、少なくとも2つのスライダーセグメントが「差分カウント」、つまり「センサー信号」値が 0 より大きい場合は、この指の静電容量値を使用してください。それ以外の場合は、上記の条件が満たされるまで指の静電容量値を減らしてください。Figure 56 に、スライダーウィジェットの指の静電容量を調整する方法を示します。

最小許容指容量を設定してもこれらの条件が満たされない場合は、**手動チューニング**を使用するか、**スライダーのデザイン**の考慮事項に従ってハードウェアを修正するか、**チューニングデバッグ FAQ** を参照してください。Figure 56 は、スライダーの指の静電容量値を設定するプロセスを説明しています。

Note: 補償 IDAC を使用することを推奨します。これは、最大 C_p を持つスライダーセグメントに対してスライダーセグメントの寄生容量の変動が大きくなるためです。

CAPSENSE™性能のチューニング

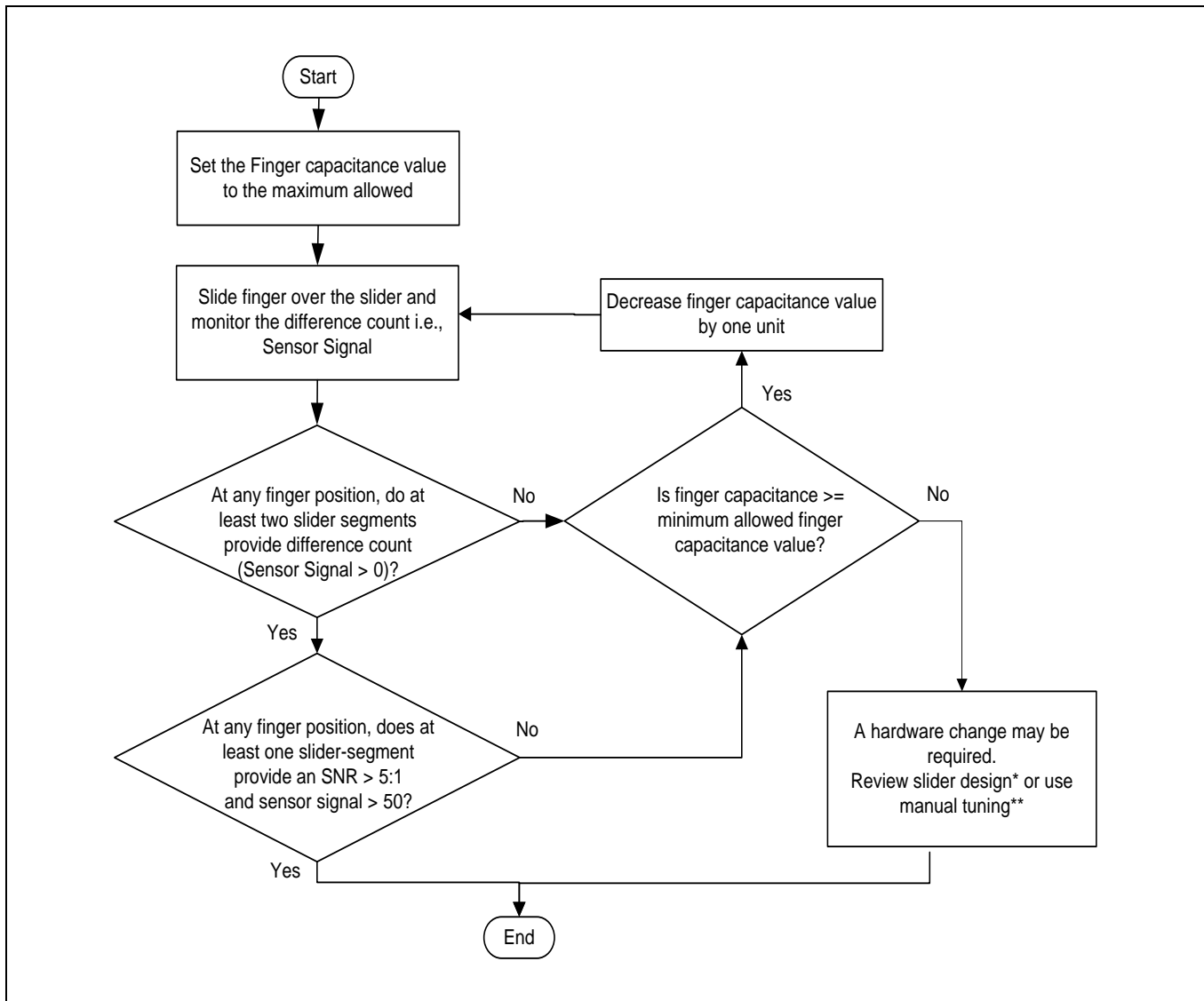


Figure 56 スライダーの指容量値の設定

* スライダーの設計を確認するために、[設計上の注意事項のスライダーの設計](#)を参照してください。

** 手動チューニングを行うために、[CAPSENSE™性能のチューニングの手動チューニング](#)を参照してください。

5.2.2.3 チューニング近接ウィジェット

[AN92239 Proximity sensing with CAPSENSE™](#)および [Getting started with CAPSENSE™ design guide](#) の「近接センシング」セクションを参照してください。

5.2.3 SmartSense ハードウェア パラメーター - Only モード

CSD チューニング方式が SmartSense (ハードウェア パラメーターのみ) のときのしきい値の推奨値については、[Table 11](#) を参照してください。

5.2.4 初期チューニングの SmartSense

詳細については、[SmartSense を使用したハードウェア パラメーターの決定](#)を参照してください。

CAPSENSE™性能のチューニング

5.3 手動チューニング

5.3.1 概要

SmartSense 技術により、デバイス自体が最適な性能を得るために校正でき、チューニング処理を完全かつ自動的に完了できます。この技術はほとんどのデザイン要件を満たしますが、SmartSense が動作しない場合、または SNR や消費電力の特定な要件がある場合、システム要件を満たすために CAPSENSE™ パラメーターを調整できます。これは手動チューニングと呼ばれます。

SmartSense の自動チューニングとは対照的に、手動チューニングはいくつかの利点があります。

- パラメーター設定による厳密な管理: SmartSense はすべてのパラメーターを自動的に設定します。しかし、パラメーターで厳密に管理する必要がある場合もあります。例えば、PSoC™ 4 がセンサーのグループをスキャンする時間または各センサーのセンスクロック周波数を厳密に管理する必要がある場合、手動チューニングを使用します (これにより、システムの EMI を低減できます)。
- より高い寄生容量に対応: 寄生容量が SmartSense の対応する値よりも高い場合は、手動チューニングを使用する必要があります。SmartSense が対応する寄生容量の範囲の詳細については、[コンポーネント データシート/ミドルウェア文書](#)を参照してください。

Figure 57 に示すように、手動チューニングのプロセスは以下の 3 つのステップに要約されます。

1. SmartSense を使用して **CAPSENSE™コンポーネント ハードウェア パラメーター**の初期値を設定するか (**SmartSense を使用したハードウェア パラメーターの決定**を参照)、手動で値を決定してください。
2. 50 カウント以上の**信号対ノイズ比**が 5:1 より大きく、システムがタイミング要件を満たすために CAPSENSE™コンポーネント ハードウェア パラメーターをチューニングしてください。
3. **CAPSENSE™ソフトウェア パラメーター**の最適な値を設定してください。

次の節では、手動チューニングの基本および上記の 3 ステップの詳細を説明します。PSoC™の CAPSENSE™アーキテクチャの知識は本節の前提条件です。PSoC™の CAPSENSE™アーキテクチャに精通するためには、[静電容量タッチセンシング方式](#)および [PSoC™ 4 および PSoC™ 6 の CAPSENSE™世代](#)を参照してください。世代間の CAPSENSE™アーキテクチャの主な違いを **Table 2** に示します。

手動チューニング手順は選択されたセンシング方式により異なります。それぞれの手動チューニング手順については [CSD センシング方式 \(第 3 および第 4 世代\)](#)、[CSX センシング方式 \(第 3 および第 4 世代\)](#)を参照してください。設計で手動チューニングを使わない場合、これらのセクションをスキップできます。**Figure 57** に一般的な手動チューニング手順を示します。

CAPSENSE™性能のチューニング

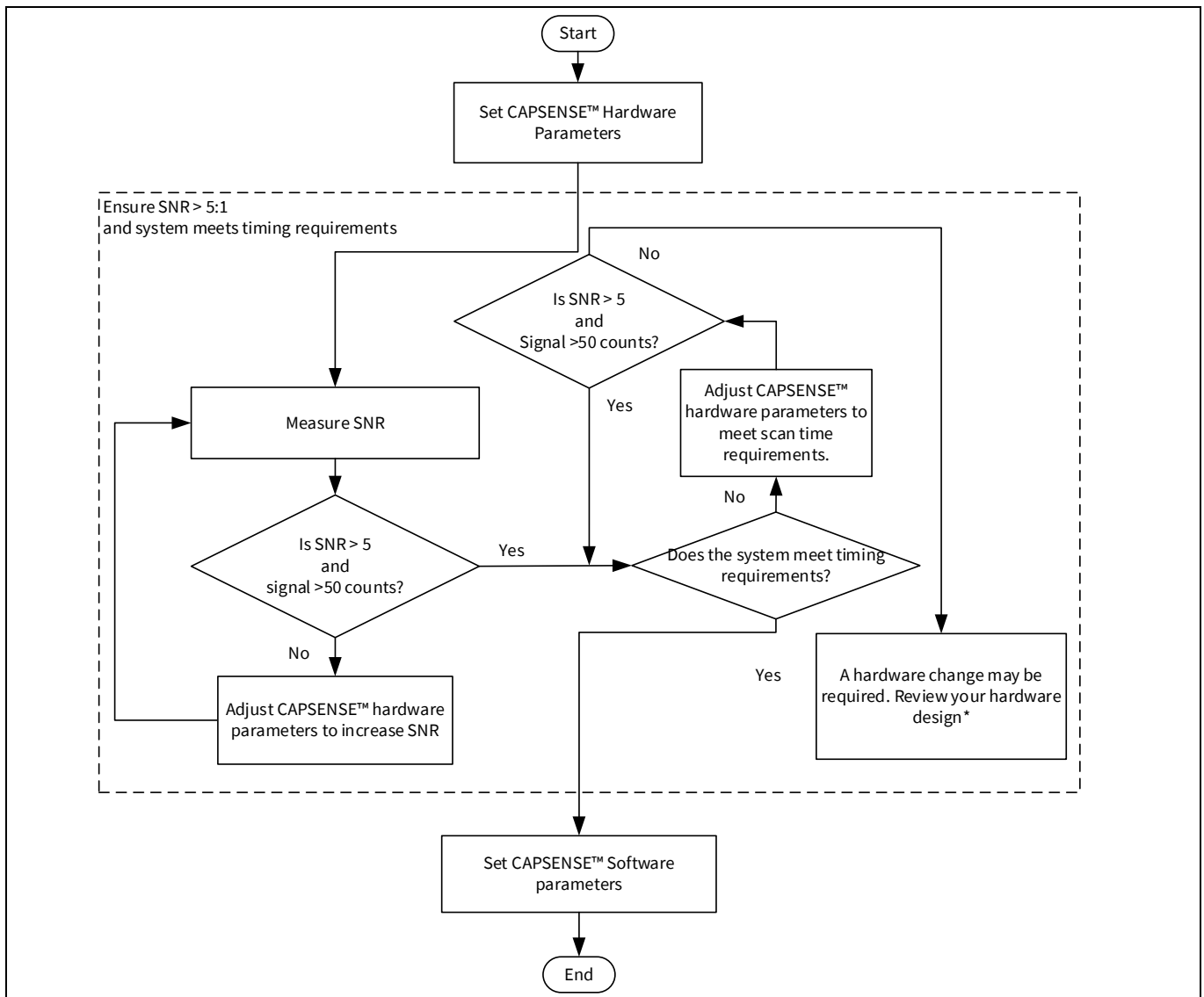


Figure 57 手動チューニングのプロセス概要

* ハードウェア設計を確認するためには、設計上の注意事項の[センサーの構造](#)および [PCB レイアウトガイドライン](#)を参照してください。

5.3.2 CSD センシング方式 (第 3 および第 4 世代)

ここでは CSD センシング方式を使用する手動チューニングの基本を説明します。また、ボタン、スライダー、タッチパッドおよび近接ウィジェットの CSD センシング方式と手動チューニング手順に影響を与えるハードウェアおよびソフトウェアのパラメーターについても説明します。

5.3.2.1 基本

5.3.2.1.1 変換利得および CAPSENSE™信号

変換利得は、センサーに指で触れたときにシステムが認識する信号の量に影響します。利得が大きい場合、信号はより大きくなり、より高い[信号対ノイズ比 \(SNR\)](#)を達成できます。利得を上げると、信号とノイズの両方が増える可能性があることに注意してください。ただし、必要に応じて、ファームウェア

CAPSENSE™性能のチューニング

フィルターを使用してノイズを減らせます。使用可能なファームウェア フィルターの詳細については、[Table 7](#) を参照してください。

シングル IDAC モードの変換利得

シングル IDAC モードでは、raw カウントはセンサーの静電容量に正比例します。

式 24. raw カウントとセンサー静電容量の関係

$$\text{raw カウント} = G_{\text{CSD}} C_S$$

ここで、

C_S = センサー容量

C_S = センサー上に指がないときの C_P

C_S = センサー上に指があるときの ($C_P + C_F$)

G_{CSD} = CAPSENSE™ CSD の静電容量-デジタル変換利得

[式 10](#) および [式 24](#) により、IDAC ソースモードを使用した変換利得の近似値は以下のとおりです。

式 25. 静電容量-デジタル変換利得

$$G_{\text{CSD}} = (2^N - 1) \frac{V_{\text{REF}} F_{\text{SW}}}{I_{\text{MOD}}}$$

式 26. 静電容量-デジタル変換利得 (IDAC ソースモード)

$$G_{\text{CSD}} = (2^N - 1) \frac{(V_{\text{DD}} - V_{\text{REF}}) F_{\text{SW}}}{I_{\text{MOD}}}$$

ここで

V_{REF} = コンパレータの基準電圧。 [Table 2](#) を参照してください。

F_{SW} = センスクロック周波数

I_{MOD} = 現在の変調器 IDAC

N = シグマ-デルタ変換器の分解能

変換ゲインの調整可能なパラメーターは、 V_{REF} , F_{SW} , I_{MOD} , および N です。 [Figure 58](#) に、raw カウントとセンサー容量のプロットを示します。

CAPSENSE™性能のチューニング

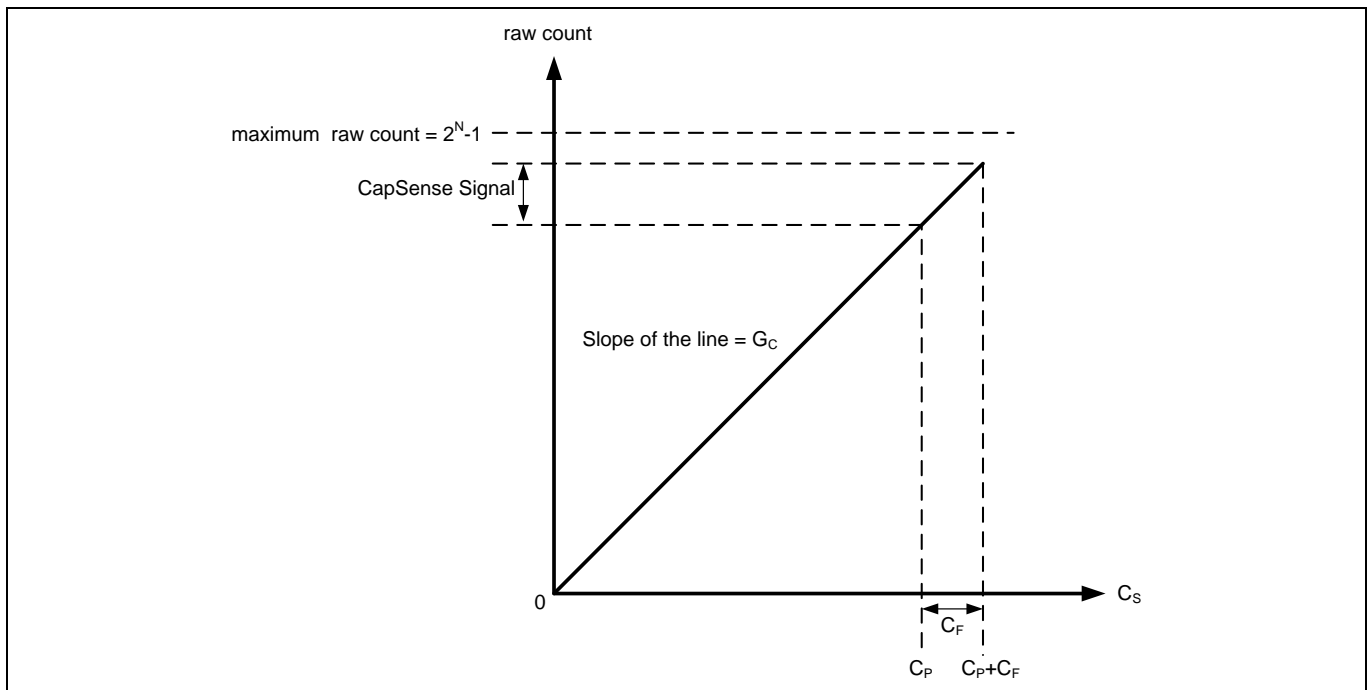


Figure 58 センサー容量に対する raw カウント

指がセンサーに触れたとき、raw カウントの変化は CAPSENSE™信号と呼ばれます。Figure 59 に変換利得に対して信号値がどのように変化するかについて示します。

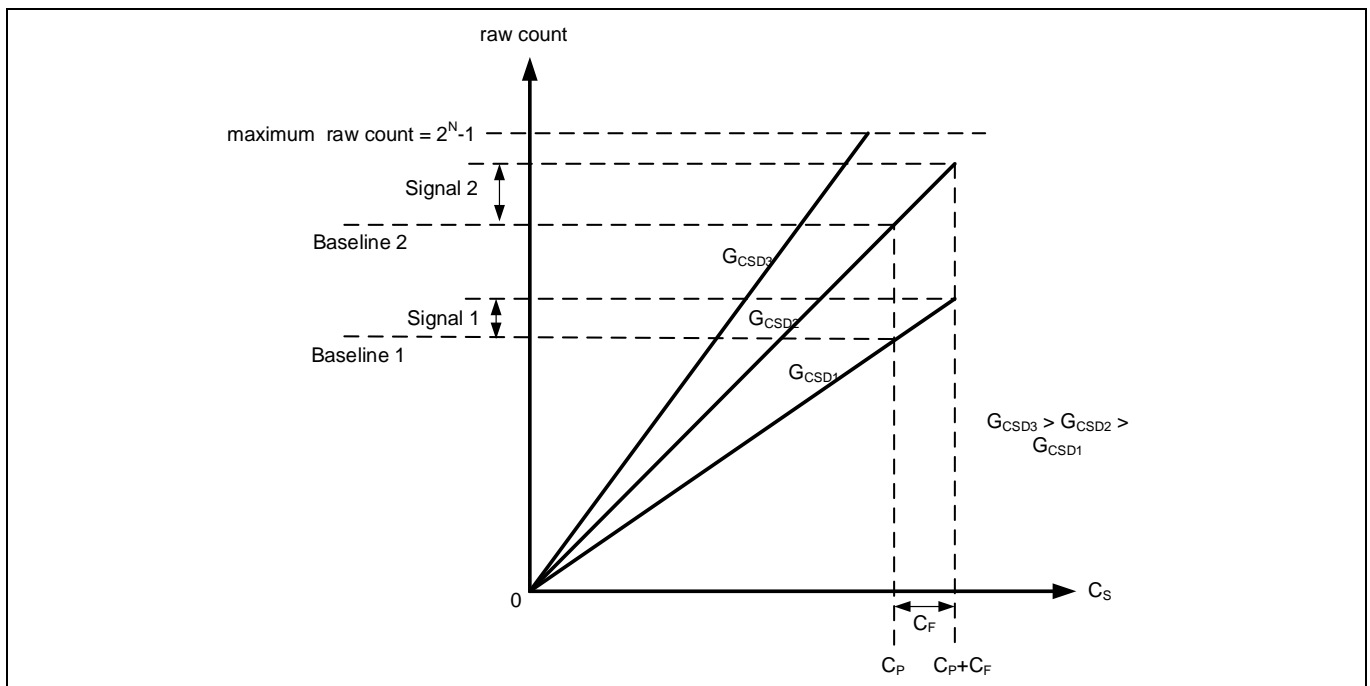


Figure 59 異なる変換利得に対する信号値

Figure 59 に3つの変換利得値 (G_{CSD3} 、 G_{CSD2} および G_{CSD1}) に該当する3つのプロットを示します。変換利得の増加により、信号値が高くなる結果となります。しかし、この変換利得の増加は、 C_p に応じた raw カウント (つまり、ベースライン) も raw カウントの最大値 (2^N-1) の方へ遷移します。非常に高い利得に

CAPSENSE™性能のチューニング

対しては、raw カウントは G_{CSD3} のプロットが示すように飽和します。そのため、raw カウントの飽和を回避しながら、良い信号値を得るために変換利得を調整する必要があります。Figure 60 に示すように、センサーに指がない場合、つまり $C_s=C_p$ の場合、raw カウントが (2^N-1) の 85% になるように、CSD パラメーターを調整してください。これにより最大利得が保証され、環境の変化によって raw カウントが増加するために十分なマージンがあり、指で触れても飽和しません。

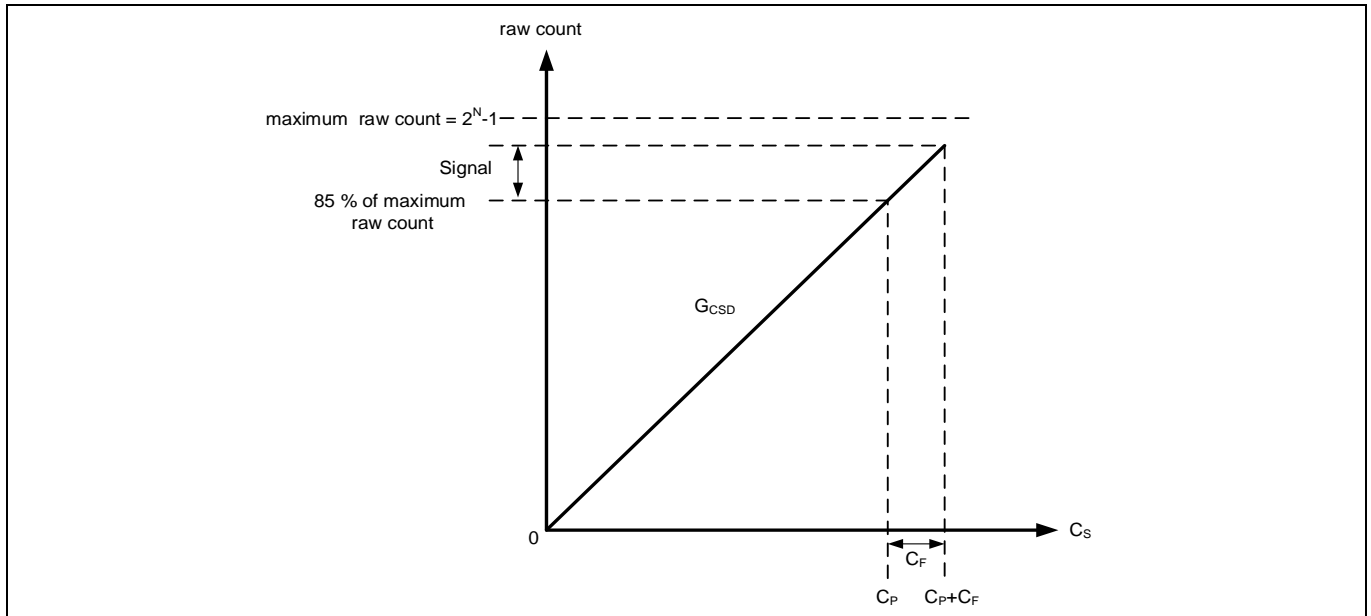


Figure 60 推奨されるチューニング

デュアル IDAC モードの変換利得

式 24 および式 12 により、デュアル IDAC モードの raw カウント用の式は以下のとおりです。

式 27. デュアル IDAC モードの raw カウント

$$\text{raw カウント} = G_{CSD} C_s - (2^N - 1) \frac{I_{COMP}}{I_{MOD}}$$

ここで、

I_{COMP} = 現在の補正 IDAC

G_{CSD} は、IDAC ソースモードの場合は式 17 で、IDAC シンクモードの場合は式 26 で与えられます。

シングル IDAC モードとデュアル IDAC モードの両方で、CSD パラメーターを調整して、センサーに指がない場合、つまり $C_s=C_p$ の場合、Figure 61 に示すように、raw カウント $= (2^N-1)$ の 85% になります。高い変換利得を確保し、フラットスポットを回避し、環境の変化による raw カウントの飽和を回避します。

CAPSENSE™性能のチューニング

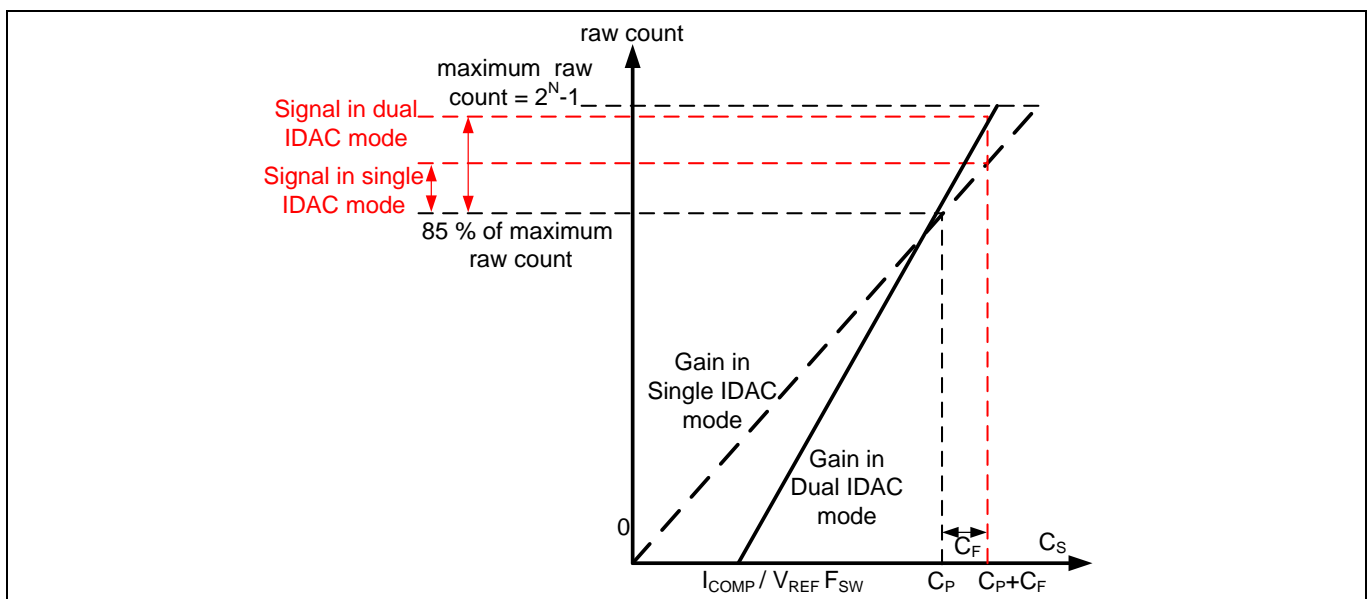


Figure 61 デュアル IDAC モードでの推奨されるチューニング

Figure 61 に示すように、85%の要件は、シングル IDAC モードでは固定ゲインに制限されますが、デュアル IDAC モードでは、 C_s 軸インターセプトを右に移動して (I_{COMP} を増やすことにより)、それに応じて変調器を減らすことにより、利得を増やせます。IDAC (I_{MOD}) は、 $C_s = C_p$ の場合でも raw カウント = $(2^N - 1)$ の 85% を達成します。このようにデュアル IDAC モードを使用すると、raw カウント対 C_p のグラフに次の変更が加えられます。

a) 補償 IDAC の使用は式 28 に示すように、 C_s 軸のゼロ以外のインターセプトを導入します。

式 28. I_{COMP} に関する C_s 軸のインターセプト

$$C_s \text{ 軸のインターセプト} = \left(\frac{I_{COMP}}{V_{REF} F_{SW}} \right)$$

デュアル IDAC モードの I_{MOD} の値は、シングル IDAC モードの I_{MOD} の値の半分であるため (残りすべてのパラメータは同じ)、デュアル IDAC モードの利得 G_{CSD} は、式 17 によりシングル IDAC モードのゲインの 2 倍になります。したがって、特定の分解能 N に対して、デュアル IDAC モードの信号はシングル IDAC モードの信号の 2 倍になります。

センサーを手動でチューニングするとき、式 17 および式 18 だけでなく、次の点も留意してください。

1. 利得が高いほど感度が向上し、システム全体の性能は良好です。しかし、利得 G_{CSD3} のプロットで Figure 59 に示すように、raw カウントを飽和させる利得を設定しないでください。したがって、シングル IDAC モードとデュアル IDAC モードの両方で、 C_p に対応する raw カウントがシングル IDAC モードで最大 raw カウントの 85% となるように利得を設定することを推奨します。
センスクロック周波数 (FSW) は慎重に設定する必要があります。式 22 に示すように、周波数は高いほど利得が高くなりますが、センサーを完全に充放電するためにできるだけ低くする必要があります。
2. 補償 IDAC を有効にすることは利得の増大に重要な役割を果たします。上記の推奨設定で、利得は倍になります。汎用アプリケーションに使用しないときは、いつも補償 IDAC を有効にします。
3. 変調 IDAC 電流を低減すると、利得が増加します。Figure 60 および Figure 61 に示すように、最高利得を達するために IDAC を調整しますが、 C_p に対応する raw カウントは温度変化などの環境の変化に十分なマージンがあることを確認してください。

CAPSENSE™性能のチューニング

4. スキャン用の分解能ビット数の増大は利得を増加させます。1ビットの分解能の増加は、システムの利得を2倍にし、**式 8**にしたがって、スキャン時間も2倍になります。スキャン時間および利得のバランスを達するために分解能の使用は必要です。

5.3.2.1.2 フラットスポット

Figure 58 および **Figure 61** に示すように、理想的に、raw カウントはセンサー静電容量とリニア関係を持つ必要があります。しかし、実際には、センサーの静電容量値の範囲でフラットスポットまたはデッドゾーンと呼ばれる非感度ゾーンがあり、ここでは **Figure 62** に示すようにシグマデルタ変調器は同じ raw カウント値を生成することがあります。この範囲はデッドゾーンまたはフラットスポットとして知られます。

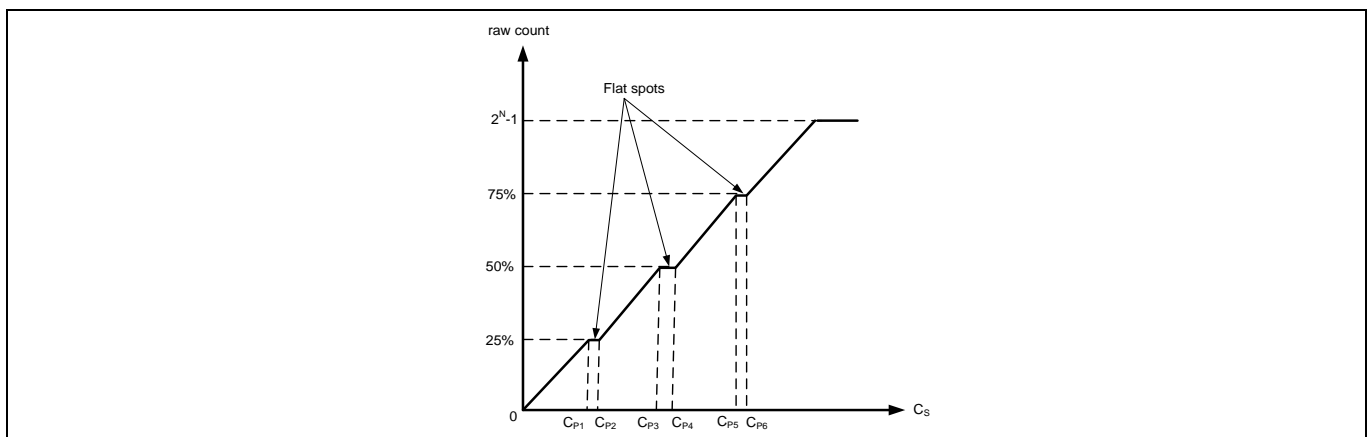


Figure 62 直接クロック使用時の raw カウント対センサー静電容量のフラットスポット

CAPSENSE™ CSD の場合、これらのフラットスポットは最大 raw カウント値の 25, 50, および 75% 近く発生します (すなわち、 2^N-1 の 25%, 50%, および 75% の近く、ここで N は **スキャン分解能**)。センスクロックソースとして直接クロックが使用されると、これらのフラットスポットは顕著になります。PRS がセンスクロックソースとして使用される場合、フラットスポットは発生しません (**SmartSense を使用したハードウェアパラメーターの決定**を参照してください)。

センスクロック に示すように、PRS はフラットスポットの影響を制限し、EMI/EMC 利点を提供するため、ほとんどのシステムに対して、センスクロックソースとして使用することを推奨します。システムは直接クロックが必要な場合、**自動校正** を使用するか、または手動校正を使用し、この raw カウントの範囲は避けることを確認してください。

フラットスポット低減技術

5. Raw カウントを 85% に調整

CAPSENSE™ CSD の場合、これらのフラットスポットは、最大 raw カウント値の 25, 50, および 75% 付近 (すなわち、 2^N-1 の 25%, 50%, および 75% の近く、ここで N は **スキャン分解能**) に発生します。キャリブレーションを 85% に設定すると、フラットスポットの幅が大幅に減少します。

6. PRS クロックの使用

これらのフラットスポットは、ダイレクトクロックが **センスクロック** ソースとして使用されている場合に顕著になります。PRS がセンスクロックソースとして使用されている場合、フラットスポットは発生しません (**SmartSense を使用したハードウェアパラメーターの決定**のセクションも参照してください)。ほとんどすべてのシステムで、PRS をセンスクロックソースとして使用することを推奨します。これは、フラットスポットの影響を制限し、**センスクロック** に示されている EMI/EMC の利点も提供するた

CAPSENSE™性能のチューニング

めです。システムで直接クロックが必要な場合は、[自動キャリブレーション](#)を使用するか、手動キャリブレーションを使用するときにこの raw カウント範囲を回避してください。

5.3.2.2 CAPSENSE™ハードウェアパラメーターの選択

CAPSENSE™ハードウェアパラメーターは変換利得および CAPSENSE™信号を支配します。[Table 8](#) に CSD センシング方式に適用する CAPSENSE™ハードウェアパラメーターの一覧を示します。次の節は CAPSENSE™ CSD システムの性能を最適化するために、これらパラメーターの調整方法を説明します。

この設計ガイドでは、説明を簡略にするために PSoC™ Creator での CAPSENSE™パラメーターの選択について説明します。同じ手順を使用して、ModusToolbox™のパラメーターを設定できます。ただし、ModusToolbox™では、分周器の値を使用してセンスクロックと変調クロックを設定しますが、PSoC™ Creator では、コンフィギュレーターで直接周波数値を指定します。CAPSENSE™の構成の詳細については、[コンポーネントデータシート/ミドルウェア文書](#)を参照してください。

Table 8 CAPSENSE™コンポーネントハードウェアパラメーター

No.	PSoC™ Creator の CAPSENSE™パラメーター	ModusToolbox™の CAPSENSE™パラメーター
1	センスクロック周波数	センスクロック分周器
2	センスクロックソース	センスクロックソース
3	変調クロック周波数	変調クロック分周器
4	変調 IDAC	変調 IDAC
5	補償 IDAC	補償 IDAC
6	スキャン分解能	スキャン分解能

5.3.2.2.1 SmartSense を使用したハードウェアパラメーターの決定

[Table 8](#) に示されているパラメーターは、CAPSENSE™ハードウェアパラメーターです。これらのパラメーターを手動で最適な値に調整することは、時間のかかる作業です。SmartSense を使用して、これらのハードウェアパラメーターを決定し、手動チューニングの初期値として使用できます。これらの値を微調整して、スキャン時間、SNR、消費電力をさらに最適化したり、CAPSENSE™システムの EMI/EMC 機能を改善したりできます。

調整モードを SmartSense に設定し、指の静電容量以外のパラメーターのデフォルト値を構成してください。調整手順は [SmartSense](#) セクションを参照してください。Tuner GUI を使用して SmartSense によって設定されたすべてのハードウェアパラメーターを読み戻してください。Tuner GUI の使用方法の詳細については、[コンポーネントデータシート/ミドルウェア文書](#)を参照してください。

[Figure 63](#) に、0.1 pF の最小指容量を検出するために、特定のハードウェア用に SmartSense によって調整された Tuner GUI の最適なハードウェアパラメーター値を示します。

CAPSENSE™性能のチューニング

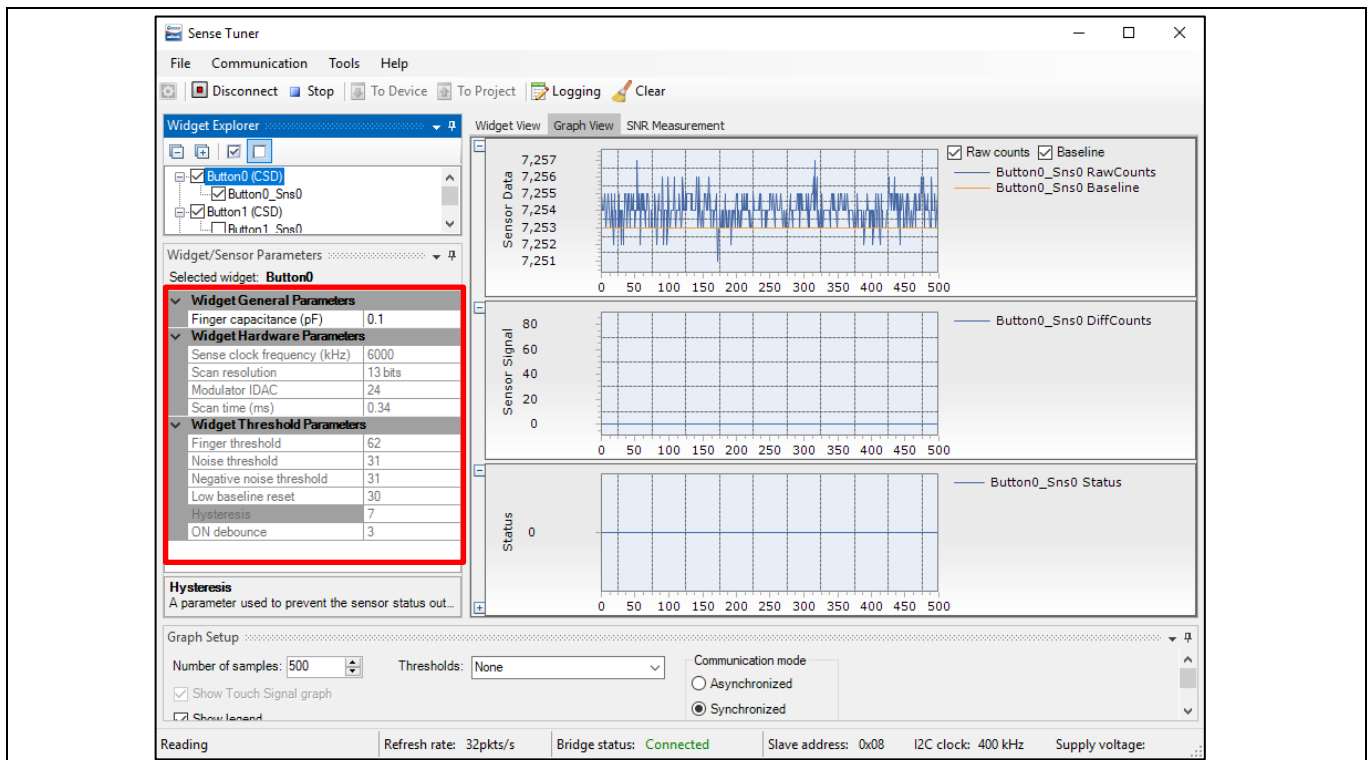


Figure 63 Tuner GUI でのハードウェアパラメーター値の読み戻し

5.3.2.2.2 ハードウェアパラメーターの手動チューニング

センスクロックパラメーター

センスクロックに関連する 2 つのパラメーターがあります。センスクロックソースとセンスクロック周波数です。

センスクロックソース

「Auto」を選択すると、コンポーネントは、ウィジェットごとに Direct, PRSx, および SSCx から最適なセンスクロックソースを自動的に選択します。Auto を選択しない場合は、以下に基づいてクロックソースを選択します。

- PRSx (疑似ランダムシーケンス) モードを使用して、フラットスポットを削除してください。
- SSCx (スペクトラム拡散クロック) モードを使用して、特定の周波数での EMI/EMC ノイズを低減してください。この機能は、PSoC™ 4 S シリーズ、PSoC™ 4100S Plus、PSoC™ 4100PS、および PSoC™ 6 ファミリのデバイスで利用できます。この場合、センスクロックの周波数は所定の範囲に広がります。
- 絶対静電容量の測定にはダイレクトクロックを使用してください。

センスクロックソースとして PRSx を選択するときは、シーケンスが 1 つの変換サイクル内に完了することを確認してください。シーケンスを完了させないと、raw カウントで高いノイズが発生する可能性があります。つまり、 $T_{PRS} \ll T_{SCAN}$ です。

PRS クロックの場合、次の式を使用して、1 つの PRS シーケンス完了サイクルとスキャン時間を計算します。

CAPSENSE™性能のチューニング

式 29. センサースキャン時間

$$T_{\text{SCAN}} = \frac{2^N - 1}{F_{\text{MOD}}}, \quad \text{ここで } N \text{ はスキャン分解能です。}$$

式 30. PRS シーケンス期間

$$T_{\text{PRS}} = \frac{2^{N_{\text{PRS}}} - 1}{F_{\text{SW}}}, \quad \text{ここで } N_{\text{PRS}} \text{ は } 8 \text{ または } 12 \text{ です。}$$

SSCx 選択のルールと推奨事項の詳細については、[コンポーネントデータシート/ミドルウェア文書](#)を参照してください。

センスクロック周波数

Figure 39 に示すように、センサーが各センスクロック期間で完全に充電および放電するように、センスクロック周波数を選択する必要があります。

これには、**式 31** に従って最大センスクロック周波数を選択する必要があります。

式 31. センスクロック最大周波数

$$F_{\text{SW}}(\text{maximum}) = \frac{1}{10R_{\text{SeriesTotal}}C_p}$$

式 32. 合計直列抵抗

$$R_{\text{SeriesTotal}} = R_{\text{EXT}} + R_{\text{GPIO}}$$

ここで、 C_p はセンサーの寄生容量です。そして $R_{\text{SeriesTotal}}$ は、内部スイッチの 500Ω の内部スイッチの抵抗、560Ω の推奨外部直列抵抗 (センサーパッドをデバイスピンに接続する PCB トレースに接続)、および高抵抗材料 (例: ITO または導電性インク) を使用する場合のトレース抵抗を含む合計直列抵抗です。つまり、合計は 1.06kΩ にトレース抵抗を加えたものです。

C_p の値は、CSD Built-in-Self-test API; `GetSensorCapacitance()` を使用して見積もれます。詳細については、[コンポーネントデータシート/ミドルウェア文書](#)を参照してください。

式 25 は、良好なゲインを得るためには最大クロック周波数を使用するのが最適であることを示します。ただし、**Figure 39** に示すように、センサーコンデンサが完全に充電および放電されることを確認する必要があります。

一般的に、シールド電極の C_p はセンサー C_p と比較して高くなります。良好な耐液性のために、シールド信号は[シールド電極調整理論](#)に記載されている条件を満たす必要があります。満たされない場合は、センスクロック周波数をさらに下げて条件を満たします。

変調クロック周波数

変調クロックは、「センサースキャン時間」とも呼ばれる静電容量からデジタルへの変換時間を制御します (**式 8**)。

変調クロック周波数が低いということは、次のことを意味します。

変換時間が長くなります (**式 22**、**式 20** を参照)

- シグマデルタコンバーターの積分時間が長いいため、raw カウントのピークツーピークノイズが低くなります
- フラットスポット**が広がります

CAPSENSE™性能のチューニング

ほとんどの場合、変換時間を最短にするためには最高周波数を選択し、フラットスポットを狭くしてください。必要に応じて、低速の変調クロックを使用して、raw カウントのピークツーピークノイズを低減してください。

変調 IDAC および補償 IDAC

CSD は、**Figure 35** に示すように、 C_{MOD} を充電する変調 IDAC と補正 IDAC の 2 つの IDAC をサポートします。これらは、静電容量からデジタルへの変換のための**デュアル IDAC モードでの変換利得**を制御します。CAPSENSE™コンポーネントでは、IDAC の以下の構成が可能です。

- 補正 IDAC の有効化または無効化
- IDAC の自動キャリブレーションの有効化または無効化
- 自動キャリブレーションが無効になっている場合の変調 IDAC および補正 IDAC の DAC コード選択

補償 IDAC

補正 IDAC を有効にすることは「デュアル IDAC」モードと呼ばれ、**デュアル IDAC モードの変換利得**で説明されているように信号が増加します。ほとんどの場合、補正 IDAC を有効にします。IDAC を他の汎用アナログ機能用に解放する場合にのみ、補正 IDAC を無効にします。

自動キャリブレーション

この機能により、ファームウェアは IDAC を自動的に調整して、必要な調整目標である 85% を達成できます。ほとんどの場合、自動キャリブレーションを有効にすることを推奨します。この機能を有効にすると、以下のようになります。

- パーツ間の C_p 変動があっても、raw カウントキャリブレーションを最大 raw カウントの 85% に修正
- **フラットスポット**を回避
- 最適な利得を自動的に選択

設計環境に大きな温度変化が含まれている場合、85% IDAC キャリブレーションレベルが高すぎ、温度の大きな変化に対して raw カウントが簡単に飽和し、SNR が低下することがあります。この場合、ファームウェアで `CapSense_CSDCalibrateWidget()` を使用して、キャリブレーションレベルを低く調整できます。

さまざまな環境条件下で CAPSENSE™を適切に機能させるために、IDAC コードが非常に低いまたは高いものを避けることを推奨します。7 ビット IDAC の場合、可能な 0~127 の範囲の 18~110 の IDAC コードを使用することを推奨します。CAPSENSE™チューナーを使用して、自動キャリブレーションされた IDAC 値がこの推奨範囲内にあることを確認できます。**式 24**、**式 25**、および**式 27**に基づいて、IDAC 値が推奨範囲外の場合は、 V_{ref} または F_{sw} を変更して、IDAC コードを適切な範囲に収められます。

リセット時に raw カウントレベルを測定して C_p の変化を検出する必要がある場合は、IDAC 自動キャリブレーションを無効にします。以下に例を示します。

- ボード間でのセンサー C_p の大きな変動またはレイアウトの問題の検出
- リセット時の指のタッチの検出
- 液面検知などの高度な CAPSENSE™メソッド (例えば、リセット時に液面ごとに異なる raw カウントレベルを設定)

DAC コードの選択

これは推奨されるアプローチではありません。ただし、これは、何らかの理由で自動キャリブレーションを無効にする場合にのみ使用できます。IDAC コードを取得するためには、最初に自動キャリブレーションを有効にして CAPSENSE™コンポーネントを構成し、他のすべてのハードウェアパラメーターを最

CAPSENSE™性能のチューニング

終調整に必要なものと同じにし、**チューナーGUI** を使用してキャリブレーションされた IDAC 値を読み戻します。次に、CAPSENSE™コンポーネントを再構成して自動キャリブレーションを無効にし、取得した IDAC コードを**チューナーGUI** から読み戻される固定 DAC コードとして使用します。

スキャン分解能

これは、**式 29** に従ってセンサーのスキャン時間、**式 24**, **式 25**, および**式 27** に従って変換ゲインを制御します。信号とスキャン時間のバランスを維持するために、スキャン解像度を選択する必要があります。

スキャン解像度が高いということは、次のことを意味します。

- **式 29** によるスキャン時間が長い
- raw カウントの SNR が高い (解像度を上げると、ノイズに対して不均衡な速度で信号が増加します)

一般に、できるだけ高い SNR を達成するように解像度を調整することを推奨します。ただし、システムが消費電力や応答時間に制約がある場合は、エンドシステムで少なくとも 5:1 の SNR を達成するために、最低の解像度を設定してください。スキャン時間または電力数の制約がある場合にのみ、スキャン解像度を 10:1 の SNR 未満に調整する必要があることに注意してください。

5.3.2.2.3 シールド電極の調整

液体耐性のために CSD センシング方法でシールド電極を有効にする場合、またはセンサーの C_p を下げる場合は、シールド関連のパラメーターを追加で構成または調整する必要があります。

シールド電極調整理論

理想的には、シールド波形は、**被駆動シールド信号およびシールド電極**で説明されているセンサーの波形とまったく同じである必要があります。ただし、実際のアプリケーションでは、シールド波形の安定時間が長くなり、オーバーシュートエラーが発生する場合があります。オシロスコープでセンサーとシールドの波形を観察してください。波形の例を**被駆動シールド信号およびシールド電極**に示します。シールド波形は、センスクロック波形のオン時間の 90%以内にセンサー電圧に落ち着く必要があります。VREF に対するシールド信号のオーバーシュートエラーは 10%未満である必要があります。

これらの条件が満たされない場合、シールドハッチに触れるとセンサーの raw カウントの変化が観察されます。さらに、**非アクティブセンサー接続**で説明したように非アクティブセンサーがシールドに接続されている場合、1つのセンサーに触れると、他のセンサーの raw カウントが変化する可能性があります。これは、シールド電極が適切に調整されていない場合にクロストークが発生することを示します。

SmartSense では、センスクロック周波数が自動的に設定されます。これらの条件が満たされているかどうかを確認してください。満たされない場合は、**手動チューニング**に切り替えて、これらの条件が満たされるようにセンスクロック周波数を手動で設定します。**シールド SW 抵抗**パラメーターを調整して、オーバーシュートエラーを減らすこともできます。

CAPSENSE™性能のチューニング

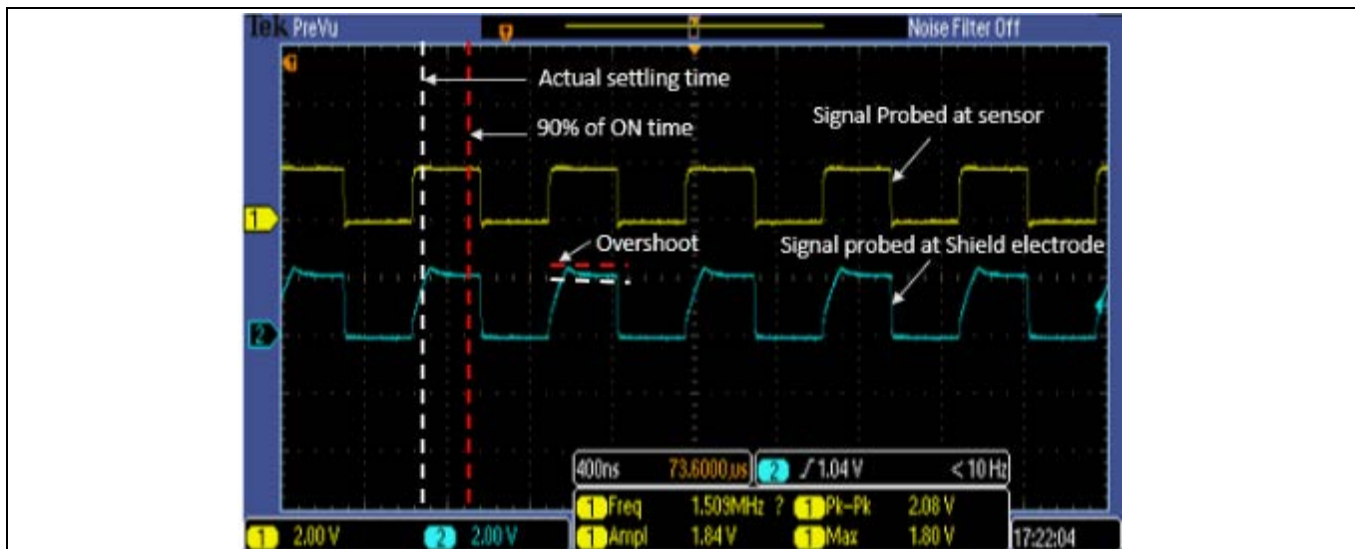


Figure 64 適切に調整されたシールド波形

シールド関連パラメーターの調整

シールドタンクコンデンサを有効にする

シールドタンクコンデンサを有効にすると、シールドの駆動強度が向上します。これによって必要に応じて早くシールド信号がセンサー電圧に安定するようになります。PSoC™ 4A-S および PSoC™ 6 MCU ファミリのデバイスにはシールドタンクコンデンサを使用することを推奨します。PSoC™ 4A、PSoC™ 4A-L、および PSoC™ 4A-M ファミリのデバイスの場合、シールドタンクコンデンサは、シールドの直列抵抗が2倍になるため、あまり有利ではありません。これらのデバイスファミリでは、このオプションを無効にしておくことを推奨します。

シールド電極の遅延

シールド電極を適切に動作させるには、シールド信号がセンサー信号と同相で一致している必要があります。センサーとシールド電極のトレース長の違いにより、シールド波形がセンサー波形より早く到着する場合があります。オシロスコープを使用してセンサー信号とシールド信号の両方を表示し、この状態を確認できます。それらが位置合わせされない場合は、このオプションを使用してシールド信号に遅延を追加し、2つの信号を位置合わせします。使用可能な遅延は、選択したデバイスによって異なります。

シールド SW 抵抗

このパラメーターは、EMI を低減するためにシールド信号の立ち上がり時間と立ち下り時間を制御します。このパラメーターは、PSoC™ 4 S シリーズ、PSoC™ 4100S Plus、PSoC™ 4100PS、および PSoC™ 6 MCU ファミリのデバイスにのみ有効です。シールドスイッチ抵抗のデフォルト値は中です。Table 9 に、シールド SW 抵抗値の影響を示します。アプリケーションの要件に基づいてこの値を選択する必要があります。さらに、シールド電極調整理論の条件を満たすことを確認してください。

CAPSENSE™性能のチューニング

Table 9 シールド SW 抵抗選択ガイドライン

より低いスイッチ抵抗	より高いスイッチ抵抗
より大きなオーバーシュートエラー	より小さなオーバーシュートエラー
より高い電磁放射	より低い電磁放射
より速い整定時間、すなわち、より高い最大セン スクロック周波数	より遅い整定時間、すなわち、より低い最大セ ンスクロック周波数

シールド電極の数

このパラメーターは、設計に必要なシールド電極の数を指定します。ほとんどの設計は、1つの専用シールド電極で機能します。ただし、一部の設計では、PCB レイアウトのルーティングを容易にするため、またはシールド層に使用される PCB の領域を最小限に抑えるために、複数の専用シールド電極が必要です。[シールド電極のレイアウトガイドライン](#)を参照してください。

非アクティブなセンサー接続

シールド電極の耐液性設計が有効な場合、またはシールドを使用してセンサーの寄生容量を低減する場合は、このオプションを「シールド」として指定する必要があります。それ以外の場合は、「グラウンド」を選択します。

ただし、非アクティブなセンサーがシールドに接続されるため、放射エミッションが高くなるリスクがあります。このような状況では、CAPSENSE™ API を使用して、非アクティブなセンサー接続を手動で制御します。未使用のすべてのセンサーをシールドに接続する代わりに、反対側の非アクティブセンサーのみまたはスキャンされるセンサーに近い非アクティブセンサーを、放射放出を減らすためにシールドに接続してください。

5.3.2.3 CAPSENSE™ソフトウェアパラメーターの選択

CAPSENSE™ソフトウェアパラメーターはセンサーの raw カウントに基づいてセンサー状態を監視します。[Table 10](#) に CAPSENSE™ソフトウェアパラメーターの一覧を示します。これらのパラメーターは CSD と CSX 両方のセンス方式に適用します。ここでは**ベースライン**を使用することで、これらのパラメーターを定義し、設計の最適な性能のために、これらのパラメーターの調整方法について説明します。[Table 11](#) に、ソフトウェアしきい値パラメーターの推奨値を示します。これらの値は、ほとんどの設計に適用できます。ただし、エンドシステムに外部ノイズが存在する場合は、センサーの誤ったトリガーを回避するために、それに応じてこれらのしきい値を変更する必要があります。

Table 10 CAPSENSE™コンポーネント ウィジェットしきい値パラメーター

No.	PSoC™ Creator/ModusToolbox™の CAPSENSE™コンポーネント パラメーター名
1.	指しきい値
2.	ノイズしきい値
3.	ヒステリシス
4.	ON デバウンス
5.	センサー自動リセット
6.	低ベースラインリセット
7.	負のノイズしきい値

CAPSENSE™性能のチューニング

Table 11 しきい値パラメーターの推奨値

Sl. No.	CAPSENSE™しきい値パラメーター	推奨値
1.	指しきい値	信号の 80%
2.	ノイズしきい値	信号の 40%
3.	ヒステリシス	信号の 10%
4.	ON デバウンス	3
5.	低ベースラインリセット	30
6.	負のノイズしきい値	信号の 40%

5.3.2.3.1 ベースライン

特定の C_p のために CAPSENSE™コンポーネントをチューニングした後、センサーの raw カウント値は、温度や湿度などの環境の変化のため徐々に変わる可能性があります。したがって、CAPSENSE™コンポーネントは raw カウントをローパス フィルターを通すことで、ベースラインと知られている新しいカウント値を作成します。**baseline** は raw カウントの緩やかな変化を追跡し、補正します。ベースラインは、指が触れることによって生じる raw カウントの急激な変化に感度が低いです。このため、ベースライン値は信号を計算するための基準レベルを提供します。Figure 65 に、raw カウント、ベースラインおよび信号の概念を示します。

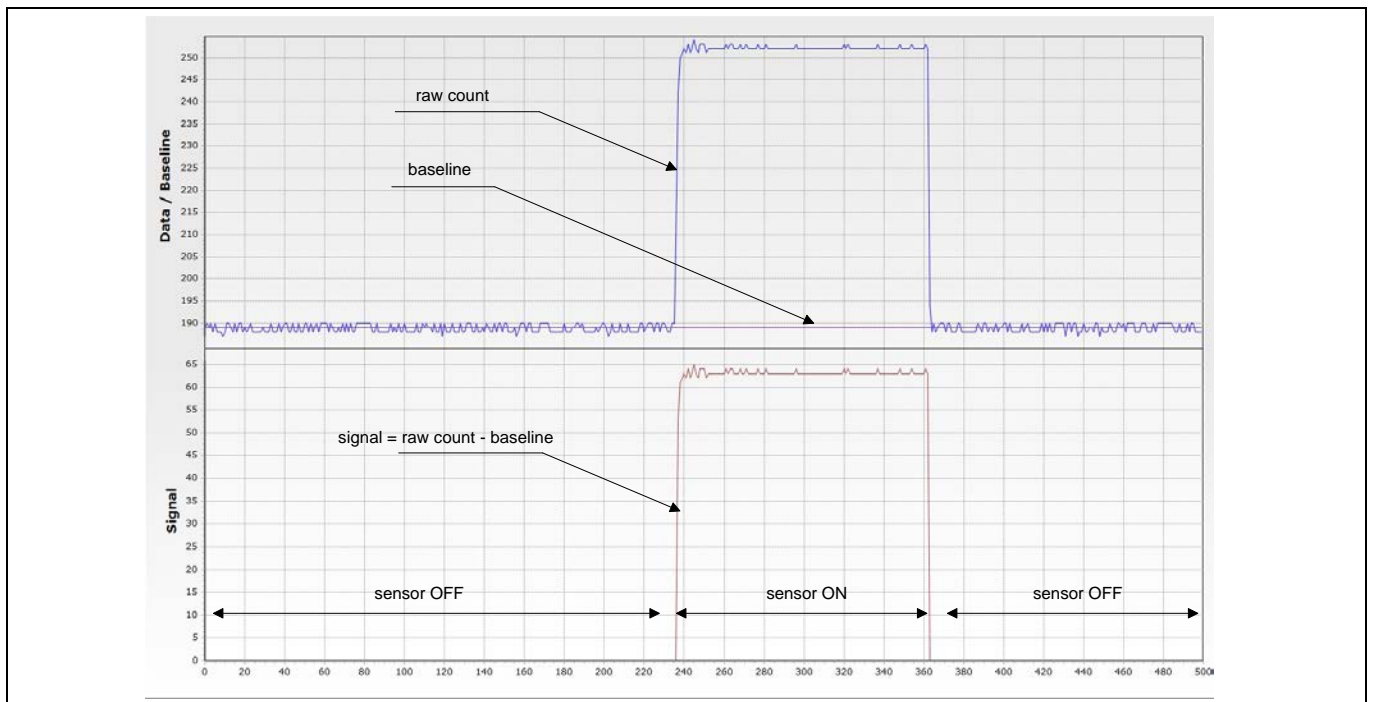


Figure 65 raw カウント、ベースライン、およびシグナル

5.3.2.3.2 ベースライン更新アルゴリズム

適切に CAPSENSE™ソフトウェア(すなわちしきい値パラメーター)を調整するために、ベースラインを計算する方法およびしきい値パラメーターがどのようにベースライン更新に影響するかを理解することが重要です。

CAPSENSE™性能のチューニング

ベースラインは raw カウントのローパスフィルターのバージョンです。Figure 66 に示すように、現在の raw カウントが (ベースライン- 負のノイズしきい値) ~ (ベースライン+ ノイズしきい値) の範囲である場合、ベースラインはローパスフィルターを通すことで raw カウントを更新します。現在の raw カウントがノイズしきい値より大きい値でベースラインよりも高い場合、ベースラインは事前のベースライン値に等しい定数値のままです。

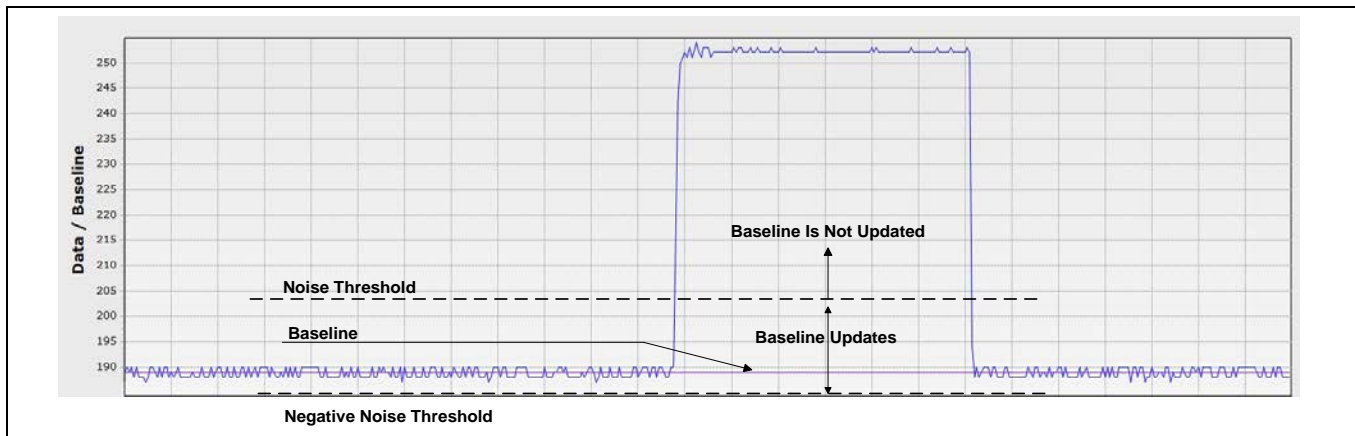


Figure 66 ベースライン更新アルゴリズム

現在の raw カウントが (ベースライン- 負ノイズしきい値) を下回る場合、ベースラインは再び、**低ベースラインリセット**のセンサー スキャン数の間、前のベースライン値と等しい値で一定のままです。Figure 67 に示すように、raw カウントが継続的に低ベースラインリセット スキャン数で (ベースライン- ノイズしきい値) よりも低いままである場合、ベースラインは現在の raw カウント値にリセットされ、再度更新され始めます。

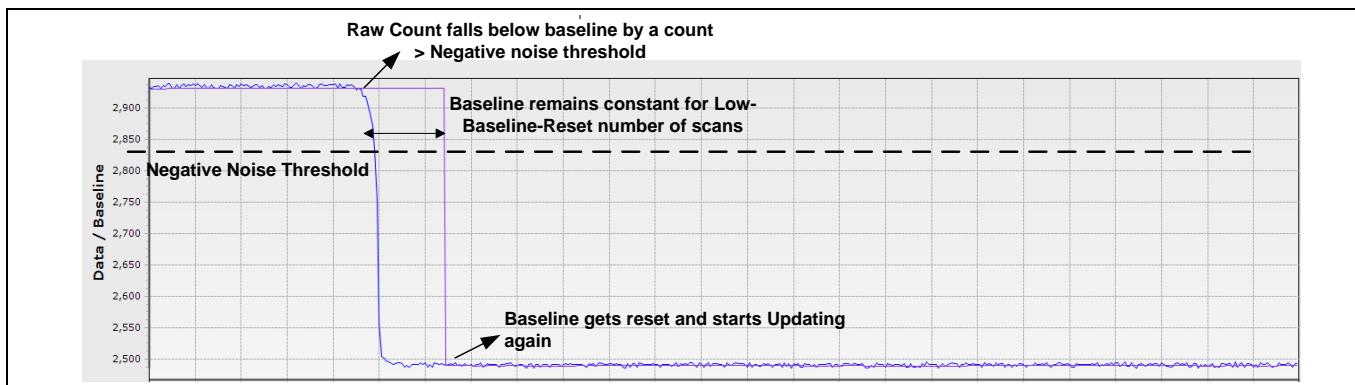


Figure 67 低ベースラインリセット

5.3.2.3.3 指しきい値

指しきい値パラメーターはヒステリシス パラメーターとともに使用され、式 33 に示すようにセンサーの状態を決定します。

式 33. センサー状態

$$\text{センサー状態} = \begin{cases} \text{オンの場合 (信号} \geq (\text{指しきい値} + \text{ヒステリシス})) \\ \text{オフの場合 (信号} \leq (\text{指しきい値} - \text{ヒステリシス})) \end{cases}$$

CAPSENSE™性能のチューニング

Figure 65 に示すように、以上の式で信号は、センサーが触れられたときの (raw カウント-ベースライン) の差を指すことに注意してください。

指しきい値は信号の 80% に設定することを推奨します。この設定は複数の PCB の信号変化で十分なマージンにセンサーのオン/オフ状態を確実に検出することを可能にします。

5.3.2.3.4 ヒステリシス

ヒステリシスパラメータは指しきい値パラメータとともに使用され、**式 33** と **Figure 68** に示すようにセンサーの状態を決定します。ヒステリシスはセンサー状態のノイズ遷移に対する耐性を提供します。ヒステリシスパラメータは、指しきい値パラメータより低く設定する必要があります。ヒステリシスを信号の 10% に設定することを推奨します。

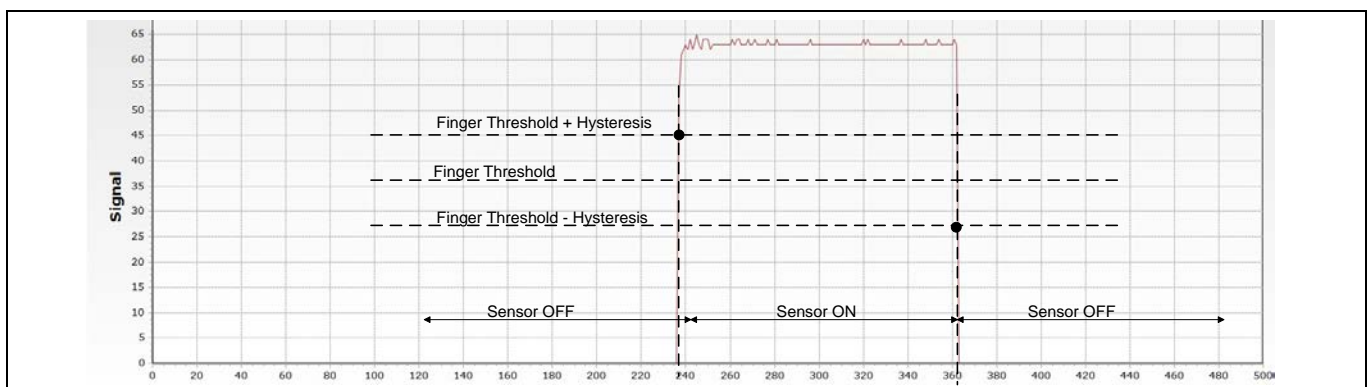


Figure 68 ヒステリシス

5.3.2.3.5 ノイズしきい値

Figure 66 に示すように、ボタンと近接センサーのようなシングルセンサーのウィジェットに対して、ノイズしきい値パラメータは raw カウントの制限値を設定します。この制限を超えると、ベースラインが更新されません。つまり、raw カウントが (ベースライン + ノイズしきい値) を超える限り、ベースラインが一定のままです。これは指のタッチの間でベースラインが raw カウントに従って変化することを防止します。

ノイズしきい値は (指しきい値 - ヒステリシス) より低い必要があります。ノイズしきい値は信号の 40% に設定することを推奨します。

ノイズしきい値を小さく設定すると、raw カウントが急に小さな値で変化してもベースラインは一定に保持され、これは電源電圧の小さな変動や高い GPIO シンク電流によるグラウンド電圧の変化があるからです。

一方、ノイズしきい値は (指しきい値 - ヒステリシス) の値の近くに設定される場合、センサーが触れられてもベースラインが更新し続けることがあります。これにより、信号が減少し (注: 信号 = raw カウント - ベースライン)、センサーの状態がオンとして報告されないことがあります。

5.3.2.3.6 負のノイズしきい値

負のノイズしきい値のパラメータは raw カウントの限界を設定し、**Figure 67** に示すように、この限界を下回るとベースラインは低ベースラインリセットパラメータで指定したサンプル数で更新されません。

CAPSENSE™性能のチューニング

負のノイズしきい値は、electrostatic discharge (ESD) イベントなどさまざまなノイズ源によって生成する raw カウントでの高振幅に繰り返しノイズスパイクがある場合、ベースラインが低くならないことを保証します。

負のノイズしきい値パラメーターをノイズしきい値パラメーターの等しい値に設定することを推奨します。

5.3.2.3.7 低ベースラインリセット

このパラメーターは負のノイズしきい値のパラメーターとともに使用されます。これは、**Figure 67** に示すように、ベースライン値をリセットするために必要となる異常に低い raw カウントを数えます。

デバイスが起動中に指がセンサーに触れた場合、ベースラインは起動時に raw カウントの高い値に初期化されます。指が離れると、raw カウントは低い値になります。この場合、ベースラインは低い raw カウントに追随すべきです。低ベースラインリセットパラメーターはこのイベントを処理するためです。これは、低いサンプルの数が低ベースラインリセット数に達するとき、ベースラインを低い raw カウント値にリセットします。この場合には、指はセンサーから離れるとき、**式 34** により得られる低ベースラインリセット時間の間センサーが指のタッチに応答しないことに注意してください。

式 34. 低ベースラインリセット時間

$$\text{低ベースラインリセット時間} = \frac{\text{低ベースラインリセットパラメーター値}}{\text{スキャン速度}}$$

低ベースラインリセットパラメーターは以下の条件を満たすように設定する必要があります。

- 低ベースラインリセットの時間は、(ESD イベントなどノイズ源に起因する) 負のノイズが続く時間より長い。
- 低ベースラインリセット時間は、デバイスの起動時にセンサー上に触れる指がセンサーから離れた後、センサーが再応答する時間と比べて短い。

低ベースラインリセットパラメーターは一般的に 30 の値に設定されます。

5.3.2.3.8 デバウンス

このパラメーターは、コンポーネントからのオン状態を生成するために、センサーを有効にする必要がある連続的な CAPSENSE™スキャンの数を選択します。デバウンスは高周波数、高振幅ノイズで誤検出をしないように確保します。

式 35. デバウンス付きのセンサー状態

$$\text{センサー状態} = \begin{cases} \text{オン (スキャン} \geq \text{デバウンス) の場合 (信号} \geq \text{(指しきい値} + \text{ヒステリシス))} \\ \text{オフの場合 (信号} \leq \text{(指しきい値} - \text{ヒステリシス))} \\ \text{オフ (スキャン} < \text{デバウンス) の場合 (信号} \geq \text{(指しきい値} + \text{ヒステリシス))} \end{cases}$$

デバウンスパラメーターは CAPSENSE™システムの応答時間に影響を与えます。指が触れると raw カウントの値が (指しきい値 + ヒステリシス) の値を上回る場合でセンサーがオンを報告するための時間は**式 36** で計算されます。

式 36. デバウンスとセンサー応答時間の関係

$$\text{センサー応答時間} = \frac{\text{デバウンス}}{\text{スキャン速度}}$$

CAPSENSE™性能のチューニング

デバウンスパラメータはセンサー状態の信頼できる検出に対して一般的に‘3’の値に設定されます。これは、エンドユーザーシステムのノイズアスペクトに基づいて上げたり下げたりできます。

5.3.2.3.9 センサー自動リセット

センサー自動リセットパラメータを有効にするとともに、信号がノイズしきい値を上回っていても下回っていても、ベースラインが常に更新されるようにします。

自動リセットを無効にすると、ベースラインは **Figure 66** および **ベースライン更新アルゴリズム** に示すように、現在の raw カウントが (ベースライン - 負のノイズしきい値) と (ベースライン + ノイズしきい値) の範囲にある場合にのみ更新されます。しかし、自動リセットを有効にすると、**Figure 69** に示すように現在の raw カウントが (ベースライン - 負のノイズしきい値) よりも高い場合にベースラインは常に更新されます。

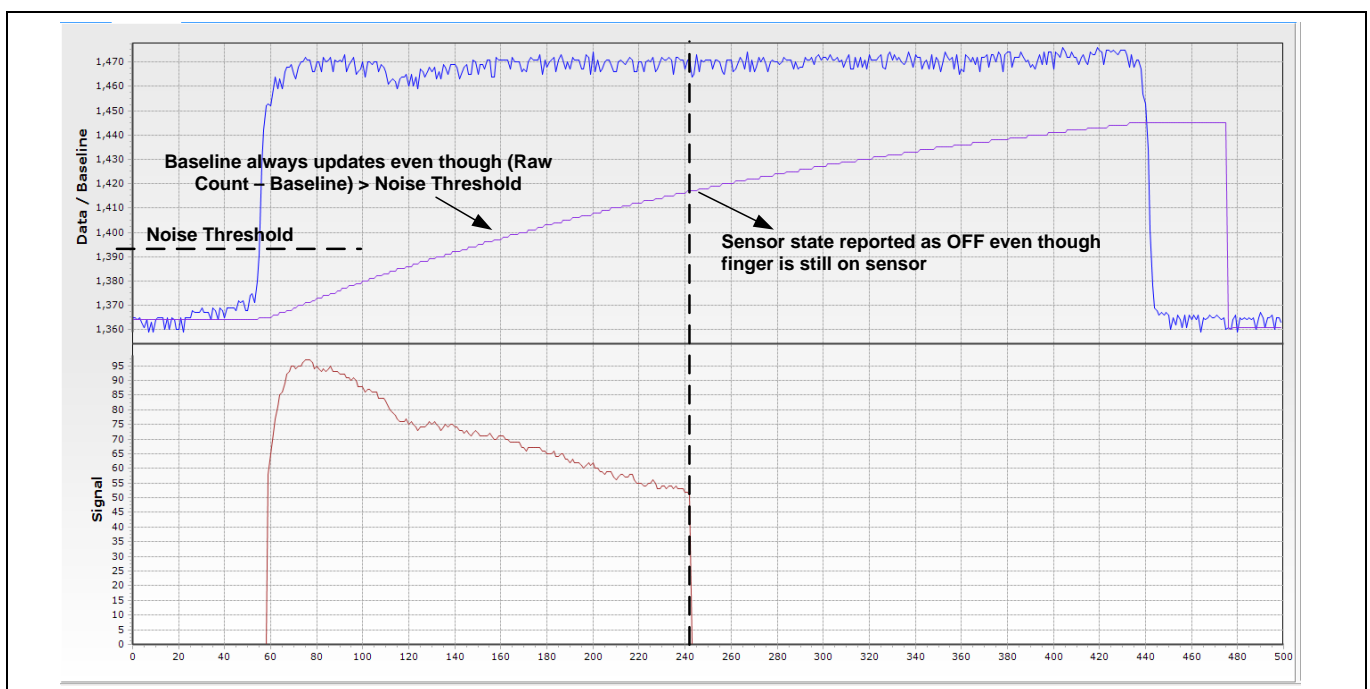


Figure 69 センサー自動リセットが有効な場合のベースライン更新

センサー自動リセットを有効にする場合、ベースラインが常に更新され、この設定はセンサーが押されたと報告される最大時間を制限します。このパラメータを有効にすることにより、何もセンサーに触れずに raw カウントが突然上昇した際に、センサーが恒久的にオンになるのを防ぎます。この突然の上昇の原因には、大幅な電源電圧の変化、高エネルギーRF ノイズ源、非常に速い温度変化があります。

何もセンサーに触れずに raw カウントが突然上昇した際に、センサーが恒久的にオンになるという問題があれば、このオプションを有効にします。

5.3.2.3.10 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。センサーの差異カウントの中央値は、さらに処理するために選択されます。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は、この機能を使用して処理を確実に行ってください。このオプションは、SmartSense

CAPSENSE™性能のチューニング

FullAutotune モードでは使用できません。サンプルコード [CE227719 CAPSENSE™ with multi-frequency scan](#) を参照してください。

5.3.2.4 ボタンウィジェット調整

Figure 70 に、CSD ボタンの調整手順の概要を示します。

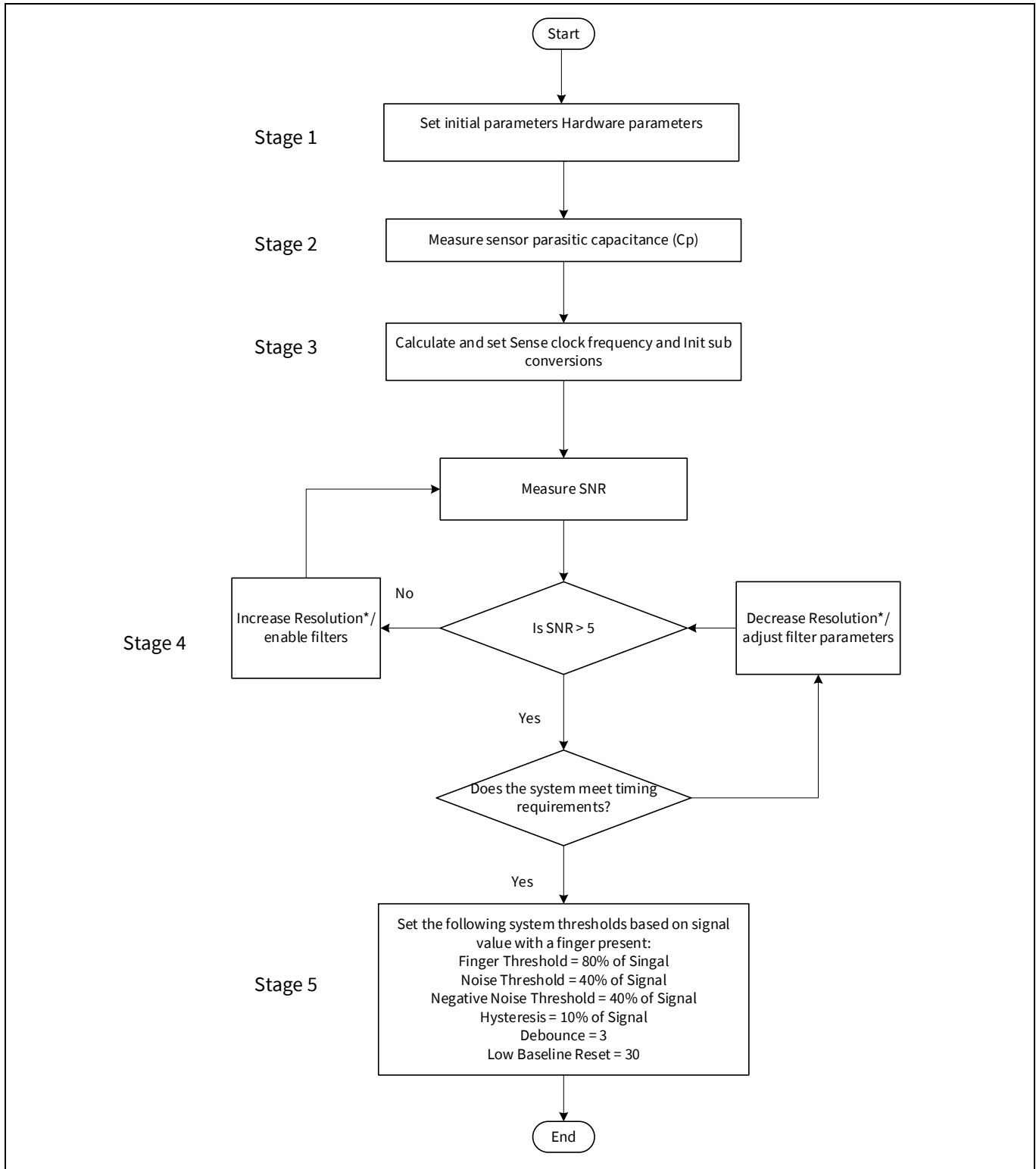


Figure 70 CSD ボタンウィジェットチューニングのフローチャート

CAPSENSE™性能のチューニング

* 第 5 世代 CAPSENSE™の場合、解像度ではなくサブ変換の数 (N_{Sub}) を変更します。

ハードウェア設計を確認するためには、設計上の注意事項の章の**センサーの構造**および **PCB レイアウトガイドライン**を参照してください。また、高度なデバッグのガイドラインについては、**チューニングデバッグ FAQ**を参照してください。

セクション 5.1 で説明したように、手動調整では最適な CAPSENSE™パラメーターを調整する必要がありますが、応答時間や消費電力などの静電容量センシングシステムの特性を厳密に制御できます。ボタンは、ノイズの多い環境での誤ったトリガーを回避するために、信頼性の高いタッチ検出用に調整されています。

CE230926 PSoC™ 4: CAPSENSE™ CSD button tuning では、CAPSENSE™ **チューナーGUI** を使用した ModusToolbox™の Eclipse IDE でのボタンウィジェットベースの自己容量ベースの調整について説明します。コンポーネントおよび関連するすべてのパラメーターの詳細については、**コンポーネントのデータシート**を参照してください。

5.3.2.5 スライダー ウィジェット調整

スライダーには多くのセグメントがあり、各セグメントは PSoC™デバイスの CAPSENSE™入力ピンに接続されます。ボタン ウィジェットセンサーの簡単なオン/オフ動作と違って、スライダー ウィジェットセンサーは連携して、指または他の導体の位置を追跡します。このため、スライダー レイアウト設計は、すべてのスライダーセグメントの C_p はできるだけ揃えるようにする必要があります。センサー間の同様の C_p 値を維持することで、チューニングの手間を最小にし、スライダー全体で均一な応答を確保できます。重心の非線形性を回避するためのスライダーレイアウト設計ガイドラインの詳細については、**スライダーの設計**を参照してください。Figure 71 に示すように、スライダーセグメントの中央に指を置いたときに、すべてのスライダーセグメントからの信号が等しくなるようにしてください。スライダーセグメントの信号が異なる場合、Figure 72 に示すように、重心は非線形になります。PSoC™ Creator と ModusToolbox™では、スライダーで指が検出されない場合、またはどのスライダーセグメントも指しきい値パラメーターより大きい差分カウント値を報告しない場合、それぞれ 0xFFFF と 0x0000 の重心が報告されることに注意してください。

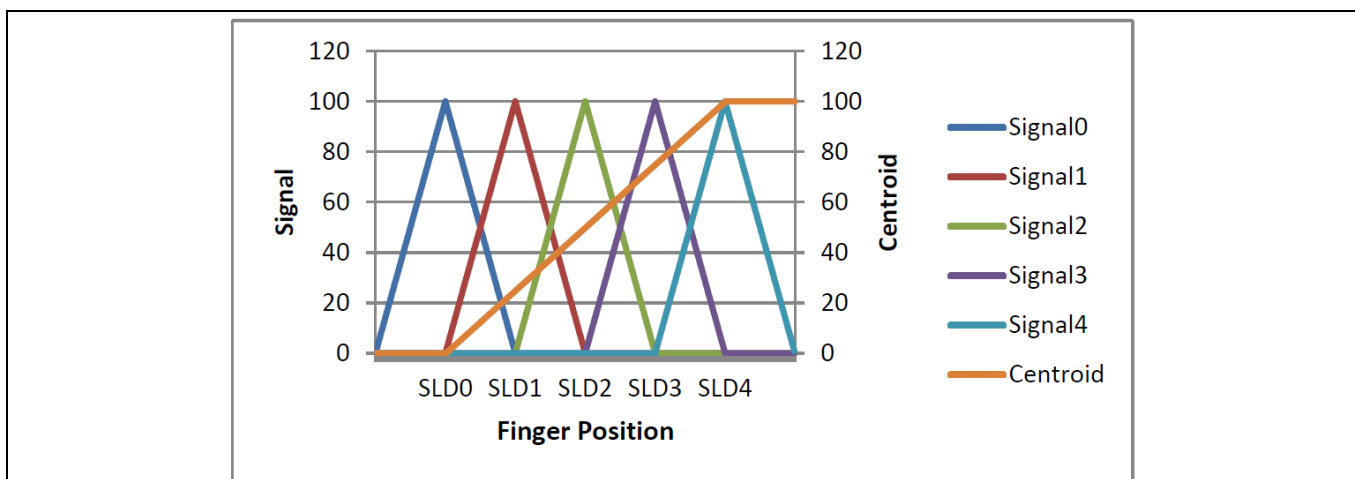


Figure 71 すべてのスライダー要素の信号が同じ場合の指位置に対する重心の対応

Note: 信号 = raw カウント - ベースライン

CAPSENSE™性能のチューニング

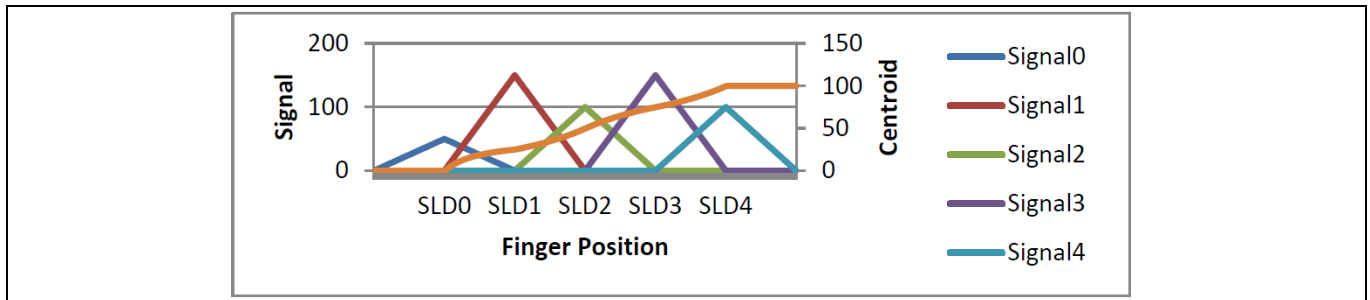


Figure 72 すべてのスライダ要素の信号が異なる場合の指位置に対する重心の応答

報告された指の位置 (つまり、重心位置) とスライダー上の実際の指の位置の線形応答では、指がセグメント SLDn の中央とセグメント SLDn-1 の中央の間のスライダーセグメントの真ん中を除いたどこかに配置されるようにし、正確に 2 つのセンサーが有効な信号⁹を報告するようにスライダーを設計する必要があります。スライダーセグメントの真ん中に指を置くと、隣接するセンサーが差分カウント=ノイズしきい値を報告する必要があります。重心位置の計算は、式 37 に示すように、指に最も近いセグメントと 2 つの隣接するセグメントに基づいているため、これらの条件が必要です。

式 37. PSoC™ Creator の CAPSENSE™コンポーネントで使用されるセントロイドアルゴリズム

$$Centroid\ position = \left(\frac{S_{x+1} - S_{x-1}}{S_{x+1} + S_x + S_{x-1}} + x \right) * \frac{Resolution}{(n - 1)}$$

ここで、

Resolution = CAPSENSE™コンポーネントカスタマイザーで設定された API 解像度

n = CAPSENSE™コンポーネントカスタマイザーのセンサー要素数

x = 最大信号を与える要素のインデックス

S_i = スライダーセグメントの様々なカウント (ノイズしきい値を差し引いたもの)

Figure 73 に、CSD スライダーの調整手順の概要を示します。

⁹ここで有効な信号とは、特定のスライダーセグメントの差分カウントがノイズしきい値以上であることを意味します。

CAPSENSE™性能のチューニング

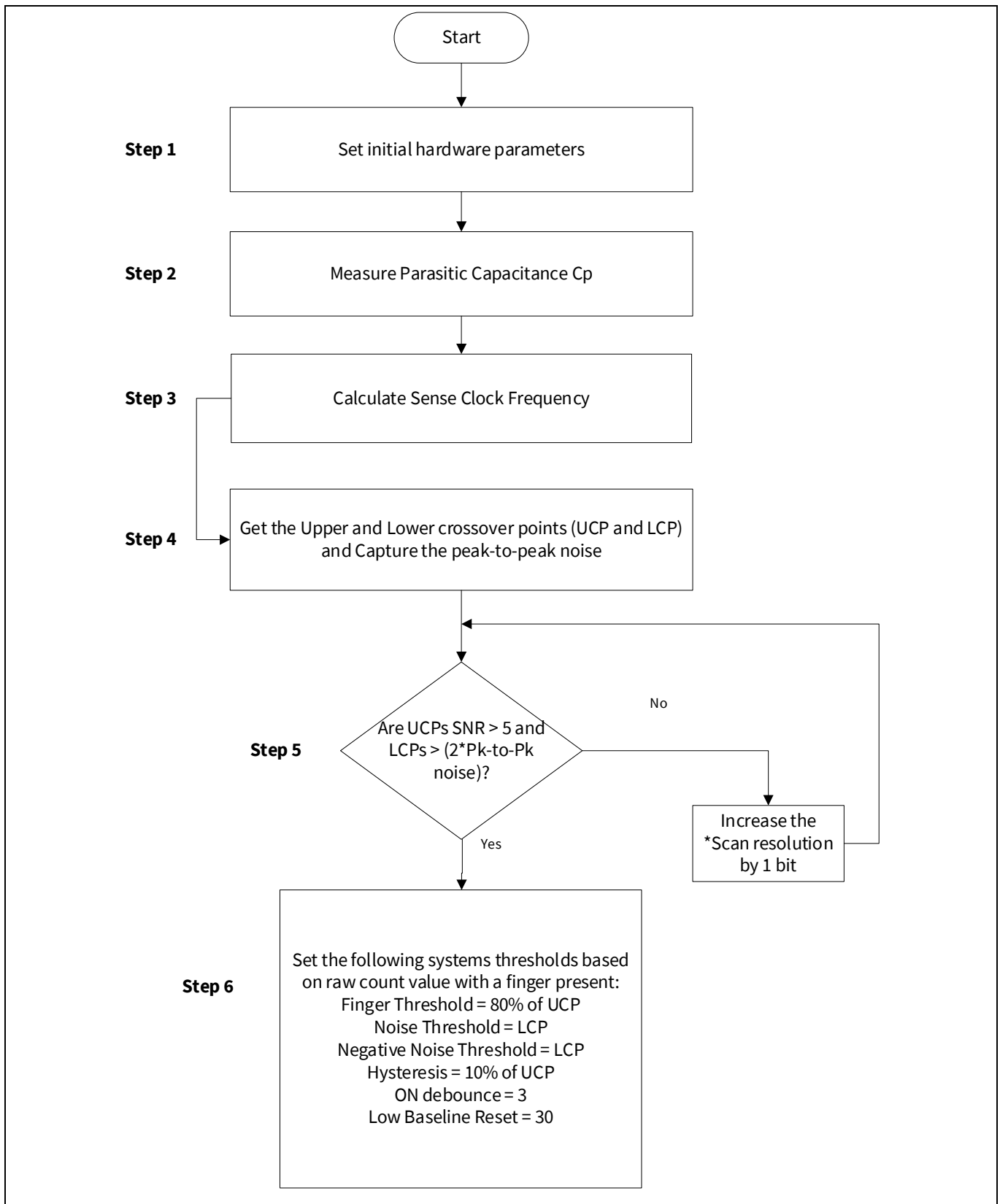


Figure 73 CSD スライダー ウィジェット チューニングのフローチャート

* 第 5 世代 CAPSENSE™ の場合、解像度ではなくサブ変換の数 (N_{Sub}) を変更します。

CAPSENSE™性能のチューニング

上部クロスオーバーポイント (UCP) と下部クロスオーバーポイント (LCP) は、**Figure 74** に示すように取得されます。PSoC™ Creator でのスライダーウィジェットベースの自己容量を手動で調整する方法を示す **CE229521 – PSoC™ 4 CAPSENSE™ CSD Slider tuning** および ModusToolbox™用の EclipseIDE での **CE230493 – PSoC™ 4: CAPSENSE™ CSD Slider tuning** を参照してください。

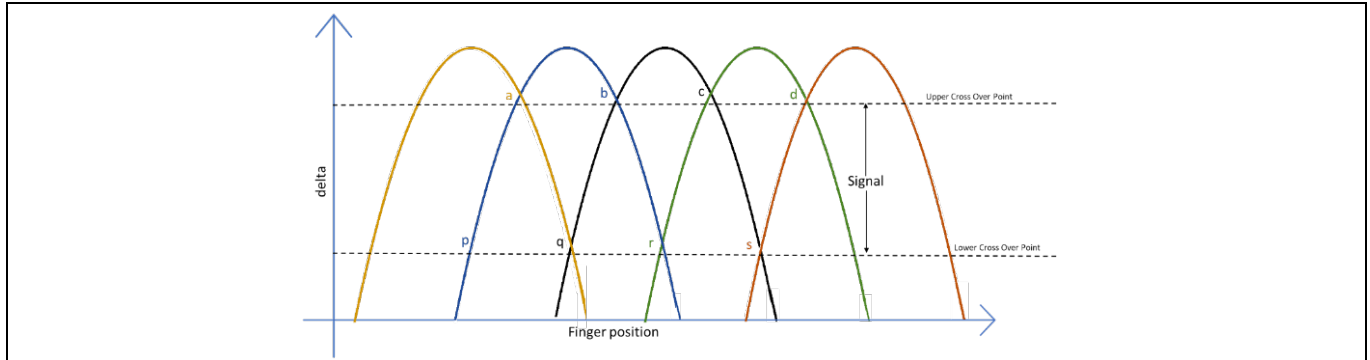


Figure 74 差分カウント (デルタ) と指の位置

5.3.2.6 タッチパッドウィジェット調整

自己容量ベースのタッチパッドは、基本的に水平方向と垂直方向に実装された2つのスライダーです。したがって、スライダーと同様の方法で調整され、トラックパッド/タッチパッド全体で均一な応答が得られます。真のマルチタッチパフォーマンスを得るには、相互容量ベースのタッチパッドを使用することを推奨します。セントロイドアルゴリズムは、すべてのセグメントから信号 (diff-counts) を取得し、x および y 位置座標を計算します。

CSD タッチパッドは、単純なスライダーとして扱われる行センサーと列センサーに個別に適用されるスライダーの重心アルゴリズムを再利用します。したがって、CSD タッチパッドの重心位置の計算式は**式 37**と同じです。

5.3.2.6.1 CSD 指検出基準

CSD タッチパッドのタッチは、次の指検出基準が満たされたときにホストに報告されます。

1. $Z_Peak > (Finger\ Threshold \pm Hysteresis)$
2. $Z_Peak > (Finger\ Threshold \pm Hysteresis) * Z3_Filt_Scale/2$ → (パネルエッジで)
3. $Z_Peak > (Finger\ Threshold \pm Hysteresis) * Z3_Filt_Scale/4$ → (パネル角で)

ここで、

Z_Peak = 指がセンサーの中心にあるときの最大信号

$Z3_sum$ = 最大信号と2つの隣接するセグメントを持つセグメントの信号の合計

$Z3_Filt_Scale = (0.8 * Z3_Sum) / \text{指しきい値}$

$Z3_Filt_Scale$ 値は、検出されたオブジェクトが正しい比率であることを保証します。

$Z3_sum$ (行と列の両方の) 条件をチェックして、指の絶対質量が指として認識されるのに十分な大きさであるかどうかを確認します。 $Z3_sum$ 条件は、ノイズによる誤接触を防ぐことができます。

CAPSENSE™性能のチューニング

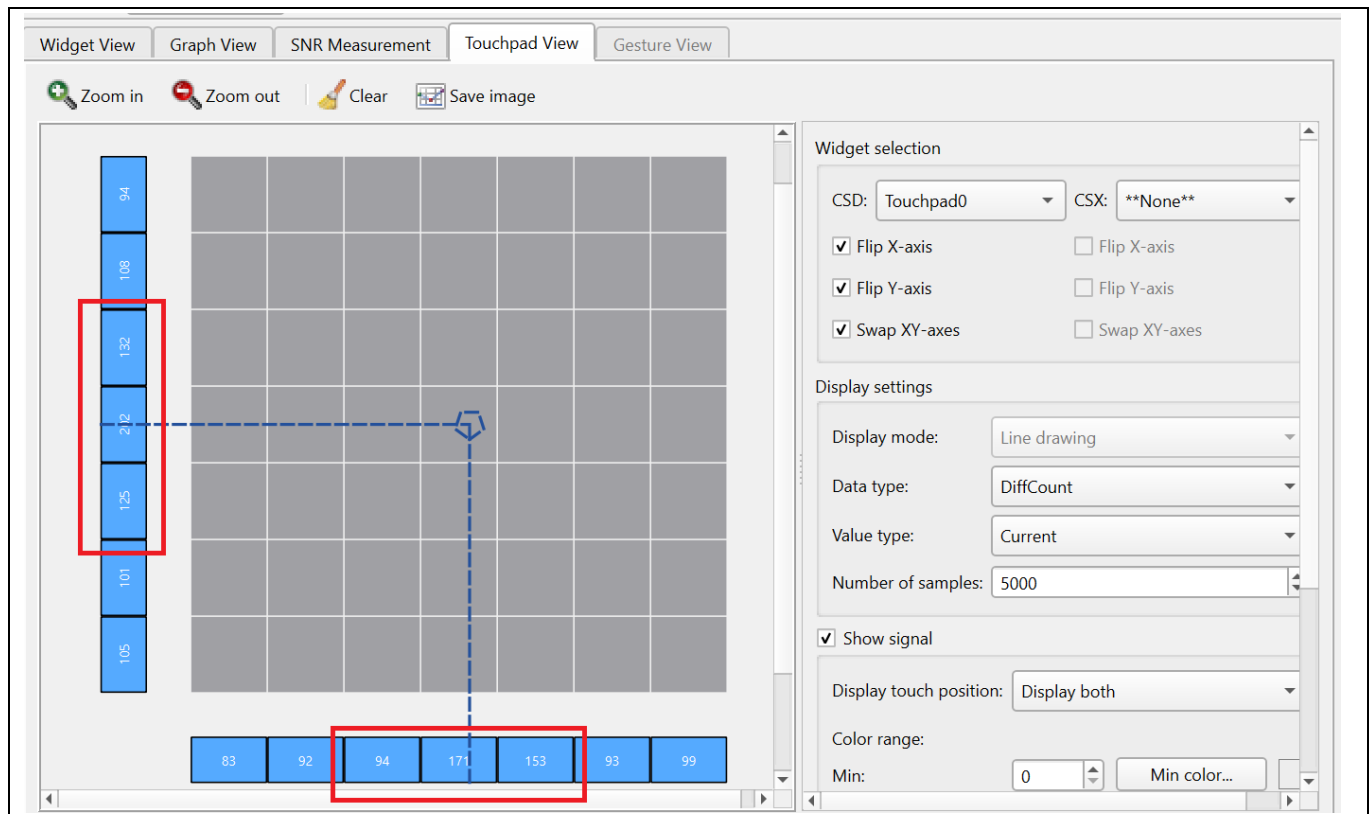


Figure 75 CAPSENSE™チューナーの Z3_sum 値

Figure 76 に、CSD タッチパッドの調整手順の概要を示します。

CAPSENSE™性能のチューニング

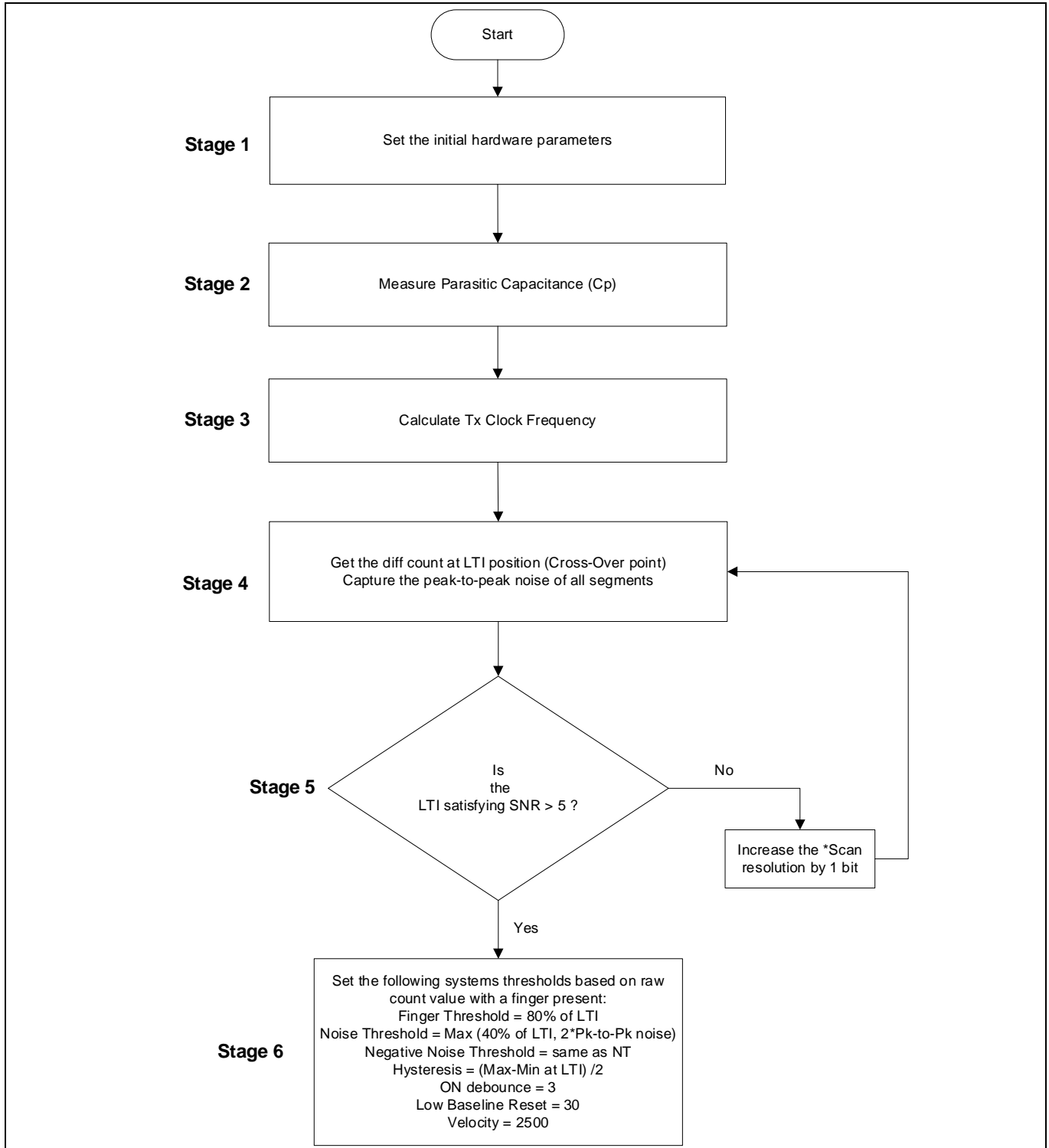


Figure 76 CSD タッチパッドウィジェットのチューニングフローチャート

*第 5 世代 CAPSENSE™の場合、解像度ではなくサブ変換の数 (N_{Sub}) を変更します。
LTI は、指で触れたときに 4 つのセンサーの中央にあるときのピーク差分カウントを測定します。LTI 信号数は、4 つのピークセンサーの平均です。これにより、有効性が最も低いタッチ信号が得られます。

5.3.2.7 近接ウィジェット調整

近接センサーの調整については、[AN92239 - Proximity sensing with CAPSENSE™](#)を参照してください。

CAPSENSE™性能のチューニング

5.3.3 CSX センシング方式 (第3 および第4 世代)

ここでは CSX センシング方式を使用する手動チューニングの基本を説明します。また、手動チューニング手順に影響を与えるハードウェアのパラメーターも説明します。

5.3.3.1 基本

5.3.3.1.1 変換利得および CAPSENSE™信号

相互容量センシングシステムでは、式 38 に示すように、 $Rawcount_{counter}$ は Tx 電極と Rx 電極間の相互容量に正比例します。

式 38. Raw カウントとセンサー静電容量の関係

$$Rawcount_{counter} = G_{CSX} C_M$$

ここで、

G_{CSX} = CAPSENSE™ CSX の静電容量-デジタルの変換利得

C_M = Tx および Rx 電極間の相互容量

Figure 78 に、CSX センサーの raw カウントと相互静電容量の関係を示します。式 39 の変換ゲインの調整可能なパラメーターは、 F_{TX} 、 N_{Sub} 、 F_{MOD} 、および IDAC です。

この変換利得の近似値は以下のとおりです。

式 39. 静電容量-デジタル変換利得

$$G_{CSX} = \frac{2 V_{TX} F_{TX} MaxCount}{IDAC}$$

式 40. 最大カウント式

$$MaxCount = \frac{F_{Mod} N_{Sub}}{F_{TX}}$$

ここで、 V_{TX} は、Figure 77 に示すように、センサーの Tx ノードの電圧です。

$$V_{TX} = V_{ON} - V_{OFF}$$

Tx クロック周波数が Tx 電極を完全に充電および放電できる場合、 V_{TX} の値は常に V_{DDIO} または V_{DDD} (V_{DDIO} が使用できない場合) です。 F_{TX} は Tx クロック周波数、 I_{DAC} は C_{INT} コンデンサを充電および放電するために引き出される電流、 N_{Sub} はサブ変換数です。

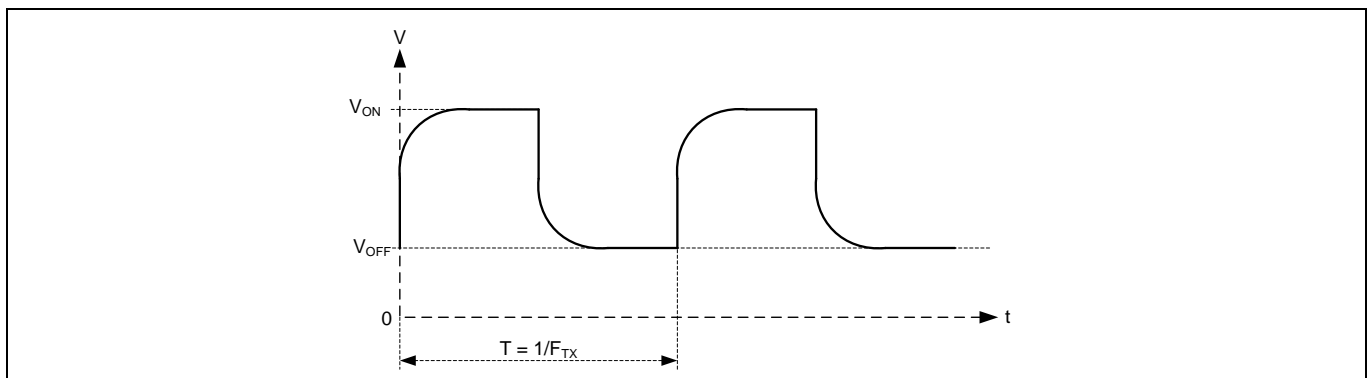


Figure 77 CSX センサーの Tx ノードでの電圧

CAPSENSE™性能のチューニング

コンポーネントから観測された raw カウントは式 41 で与えられることに注意してください。Rawcount_{component} の詳細については、CAPSENSE™ CSX センシング方式 (第 3 および第 4 世代) を参照してください。

式 41. Rawcount_{Component}

$$Rawcount_{Component} = MaxCount - Rawcount_{counter}$$

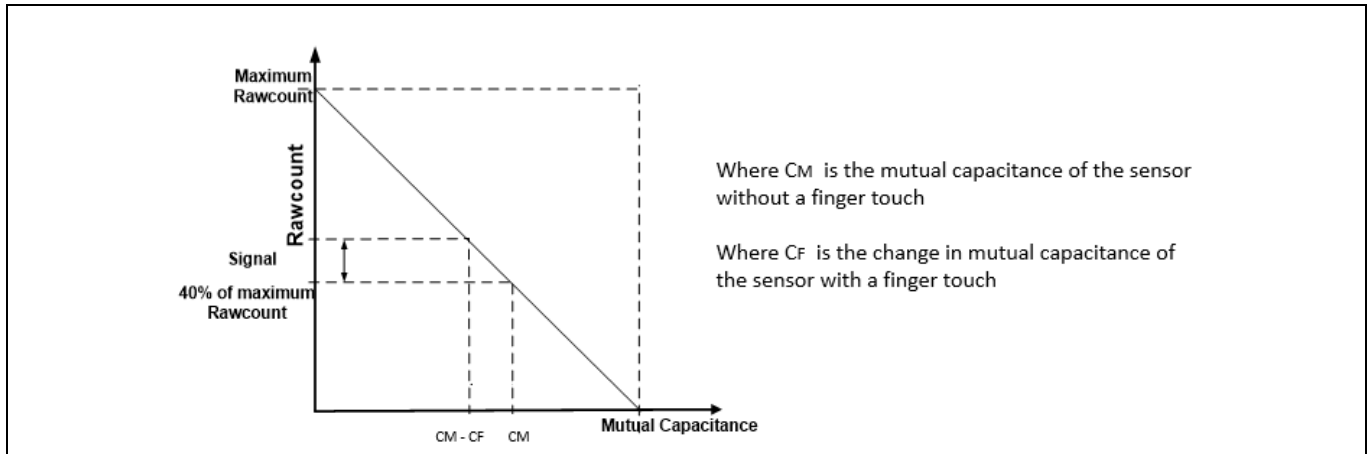


Figure 78 Raw カウントとセンサー相互静電容量

5.3.3.2 CAPSENSE™ハードウェアパラメーターの選択

CAPSENSE™ハードウェアパラメーターは、変換利得および CAPSENSE™信号を支配します。Table 12 に CSX センシング方式に対応する CAPSENSE™ハードウェアのパラメーターの一覧を示します。Table 12 に、PSoC™ Creator CAPSENSE™コンポーネントの各パラメーターの ModusToolbox™ミドルウェアのパラメーターへのマッピングも示します。ドキュメントを簡単にするために、この設計ガイドでは、PSoC™ Creator の CAPSENSE™ Configurator を使用して CAPSENSE™パラメーターを選択する方法を示します。ModusToolbox™で CAPSENSE™を構成する場合も同じ手順です。ただし、ModusToolbox™では、分周器の値を使用して Tx クロックと変調クロックを設定します。一方、PSoC™ Creator では、周波数値をコンフィギュレーターで直接指定します。コンポーネントデータシート/ミドルウェア文書を参照してください。

Table 12 CAPSENSE™信号コンポーネントハードウェアパラメーター

No.	PSoC™ Creator の CAPSENSE™パラメーター	ModusToolbox™の CAPSENSE™パラメーター
1	変調クロック周波数	変調クロック分周器
2	Tx クロック ソース	Tx クロック ソース
3	Tx クロック周波数	Tx クロック分周器
4	IDAC	IDAC
5	サブ変換回数	サブ変換回数

5.3.3.2.1 Tx クロック パラメーター

Tx クロックに関連する 2 つのパラメーターがあります。センスクロックソースとセンスクロック周波数です。

CAPSENSE™性能のチューニング

Tx クロックソース

「Auto」を選択して、コンポーネントを、ウィジェットごとにダイレクトおよびスペクトラム拡散クロック (SSCx) の間で最適な Tx クロックソースを自動的に選択してください。Auto を選択しない場合は、以下に基づいてクロックソースを選択してください。

- ダイレクト - 固定クロック周波数のクロック信号。

ほとんどの場合、このオプションを使用してください。

- スペクトラム拡散クロック (SSCx) - このオプションを選択した場合、Tx クロック信号の周波数は所定の範囲に動的に拡散されます。このオプションを使用して、EMI 干渉を減らし、フラットスポットを回避します。

ただし、SSCx クロックを選択するときは、SSCx クロックソース選択の [コンポーネントデータシート / ModusToolbox™ CAPSENSE™ configurator guide](#) に記載されている条件が満たされるように、Tx クロック周波数、変調器クロック周波数、およびサブ変換数を選択する必要があります。

Tx クロック周波数

CAPSENSE™ CSX センシング方式 (第 3 および第 4 世代) で説明されるように、Tx クロック周波数は各サブ変換時間を決めます。Tx クロック信号は、センサーの寄生容量を完全に充電および放電する必要があります。オシロスコープで信号をチェックすることで検証するか、**式 42** を使用して設定できます。さらに、自動調整された IDAC コードが、選択した F_{TX} の中間範囲 (例えば、30~90) にあることを確認する必要があります。自動調整された IDAC コードが推奨範囲外にある場合は、IDAC が推奨範囲内に収まり、**式 42** を満たすように F_{TX} を調整します。

式 42. Tx クロック周波数を選択するための条件

$$F_{TX} < \frac{1}{10R_{SeriesTx}C_{PTx}}$$

式 43 に示すように、スキャン時間を最小限に抑えるためには、この基準を満たすコンポーネントドロップダウンリストで使用可能な最大 Tx クロック周波数を使用することを推奨します。

式 43. CSX センサーのスキャン時間

$$T_{CSX} = \frac{N_{Sub}}{F_{TX}}$$

ここで、 N_{Sub} = **サブ変換回数** です。

さらに、SSCx クロックソースを使用している場合は、これらの条件に加えて、[コンポーネントデータシート / ミドルウェア文書 / ModusToolbox™ CAPSENSE™ configurator guide](#) に記載されている条件を満たす Tx クロック周波数を選択していることを確認してください。

F_{TX} の最大値は、選択したデバイスによって異なります。PSoC™ 4 S シリーズ、PSoC™ 4100S Plus、PSoC™ 4100PS、および PSoC™ 6 MCU ファミリのデバイスの場合、最大 F_{TX} は 3000kHz であり、その他のデバイスの場合は 300kHz です。

5.3.3.2.2 変調クロック周波数

式 30、**式 31** に示すように、変調クロック周波数が高いほど、特定の C_M からデジタルカウントへの変換で感度/信号が高くなり、精度が高くなり、ノイズが低くなるため、特定のデバイスで許容される最高のクロック周波数を選択するのが最適です。また、 F_{mod}/F_{tx} の値を高くすると、 C_M から raw カウントへの変換で **フラットスポット** の幅が狭くなります。

CAPSENSE™性能のチューニング

5.3.3.2.3 IDAC

IDAC 自動キャリブレーションを有効にすることを推奨します。非常に高い IDAC コードと非常に低い IDAC コードは避けるのが最善です。推奨される IDAC コードの範囲は 30~90 です。IDAC 値が推奨範囲から外れている場合は、Tx クロック周波数を調整して IDAC レベルを調整してください。IDAC が適切にキャリブレーションできない場合は、設計の C_M が低いことが原因である可能性があります。設計における低 C_M の影響を軽減するために、**CSX ボタンの C_M が低いことを確認します**セクションを参照してください。

5.3.3.2.4 サブ変換数

サブ変換数によって、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。固定変調器クロックと Tx クロックの**式 14** から、サブ変換の数を増やすと (N_{Sub})、信号と SNR が増加します。ただし、サブ変換の数を増やすと、**式 44** に従ってセンサーのスキャン時間も長くなります。

式 44. CSX スキャン時間

$$\text{スキャン時間} = \frac{N_{Sub}}{F_{TX}}$$

最初に、値を低い数値 (例えば、20) に設定し、チューナー GUI を使用してセンサーの SNR を見つけてください。選択した N_{Sub} で SNR が 5:1 を超えていない場合は、SNR 要件が満たされるように、 N_{Sub} を段階的に増やしてみてください。

5.3.3.3 CAPSENSE™ソフトウェアパラメーターの選択

相互容量の CAPSENSE™ソフトウェアパラメーターは自己容量のパラメーターと同じです。したがって、これらのパラメーターは**CAPSENSE™ソフトウェアパラメーターの選択**で説明されているように選択できます。

5.3.3.4 ボタンウィジェット調整

Figure 79 に、CSX ボタンの調整手順の概要を示します。

CAPSENSE™性能のチューニング

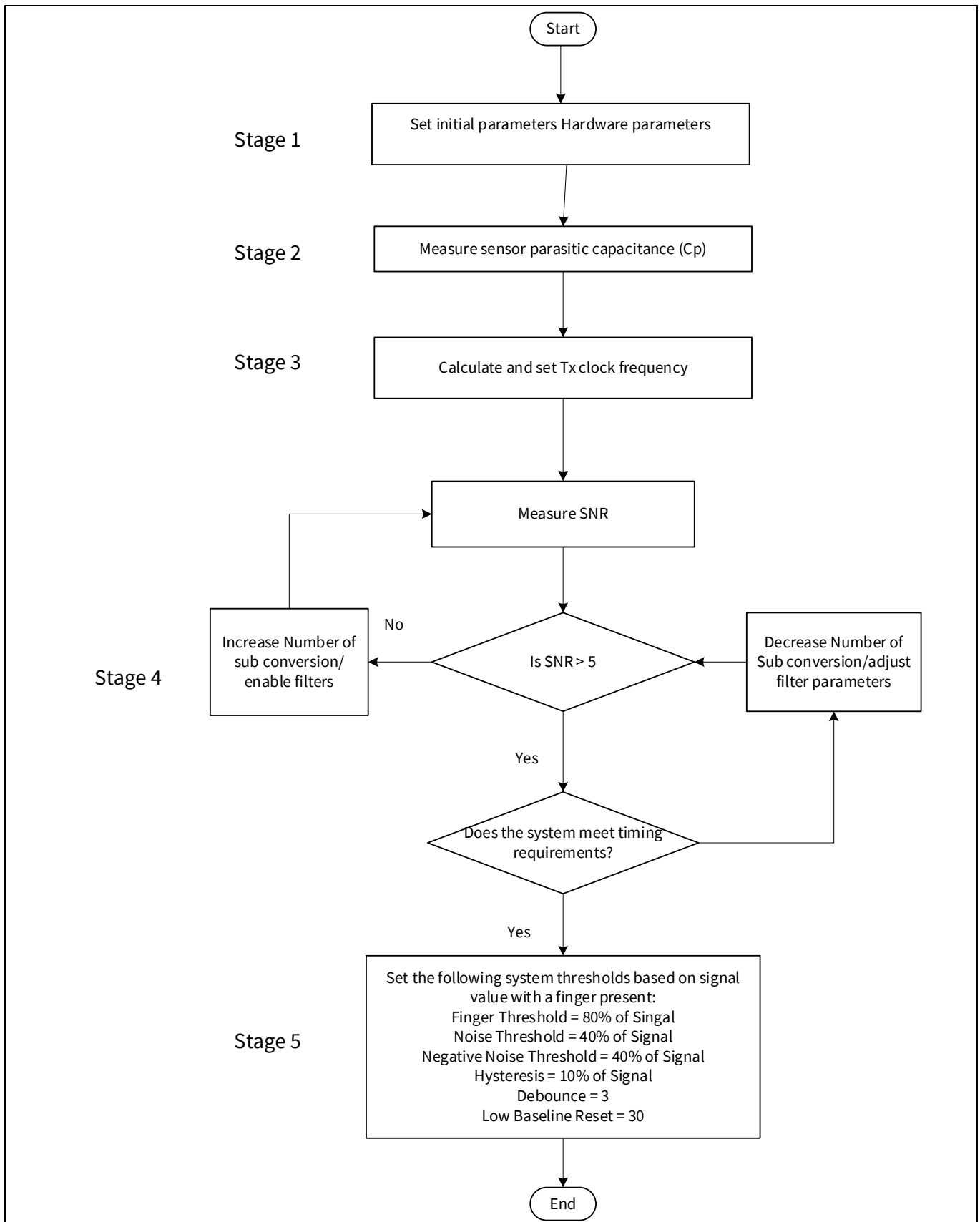


Figure 79 CSX ボタン ウィジェットチューニング例

CAPSENSE™性能のチューニング

* ハードウェア設計を確認するためには、設計上の注意事項の[センサーの構造](#)および [PCB レイアウトガイドライン](#)を参照してください。また、高度なデバッグのガイドラインについては、[チューニングデバッグ FAQ](#)を参照してください。

CE230660 PSoC™ 4: CAPSENSE™ CSX button tuning は Eclipse IDE for ModusToolbox™での相互容量ベースのボタンウィジェットのチューニングについて、および **CE228931 - PSoC™ 4 CAPSENSE™ CSX button tuning** は CAPSENSE™チューナーを使用した PSoC™ Creator について説明します。コンポーネントおよび関連するすべてのパラメーターの詳細については、[コンポーネントのデータシート](#)を参照してください。

5.3.3.5 タッチパッドウィジェットの調整

相互容量ベースのタッチパッドウィジェットは、最大3回の同時指タッチをサポートします。CSD タッチパッドとはわずかに異なる重心アルゴリズムが、CSX タッチパッドウィジェットに適用されます。3x3 アルゴリズムは、それぞれ**式 45**と**式 46**に示すように、重心アルゴリズムを使用して X 位置と Y 位置を計算するために使用されます。

式 45. CSX タッチパッドで重心アルゴリズムを使用した X 位置の計算

$$positionX = \left(\frac{S_{x+1} - S_{x-1}}{S_{3x3}} + x \right) * \frac{ResolutionX}{(n_x - 1)}$$

ここで、

ResolutionX = 最大 X 軸位置

n_x = X 方向のセンサー要素数

x = 最大信号を与える要素のインデックス

S_{x+1} = 最大値 (x) から左側にある 3 つの隣接要素の合計

S_{x-1} = 最大値 (x) から右側にある 3 つの隣接要素の合計

S_{3x3} = 3x3 差分配列の総合計

式 46. CSX タッチパッドで重心アルゴリズムを使用した Y 位置の計算

$$positionY = \left(\frac{S_{y+1} - S_{y-1}}{S_{3x3}} + y \right) * \frac{ResolutionY}{(n_y - 1)}$$

ここで、

ResolutionY = 最大 Y 軸位置

n_y = Y 方向のセンサー要素数

y = 最大信号を与える要素のインデックス

S_{y+1} = 最大値 (y) から上にある 3 つの隣接要素の合計

S_{y-1} = 最大値 (y) から下にある 3 つの隣接要素の合計

5.3.3.5.1 CSX 指検出基準

次の指検出基準が満たされると、CSX タッチパッドのタッチがホストに報告されます。

1. $Z_Peak > \text{指しきい値} \pm \text{ヒステリシス}$

CAPSENSE™性能のチューニング

2. Z9_Sum 条件

- $Z9_Sum > ((\text{指しきい値} + \text{ヒステリシス}) * Z9_Filt_Scale)$ (パネルコアで)
- $Z9_Sum > ((\text{指しきい値} + \text{ヒステリシス}) * Z9_Filt_Scale/2)$ (パネルエッジで)
- $Z9_Sum > ((\text{指しきい値} + \text{ヒステリシス}) * Z9_Filt_Scale/4)$ (パネル角で)

3. Z8_sum 条件

- $Z8_sum > Z_peak * Z8_Filt_Scale$ (パネルコアで)
- $Z8_sum > Z_peak * Z8_Filt_Scale/2$ (パネルエッジで)
- $Z8_sum > Z_peak * Z8_Filt_Scale/4$ (パネル角で)

ここで、

Z_peak = 得られた最大信号

$Z9_sum$ = 3x3 差分配列の合計

$Z8_sum$ = $Z9_Sum - Z_peak$

$Z9_Filt_Scale$ = $(0.8 * Z9_Sum) / \text{指しきい値}$

$Z8_Filt_Scale$ = $(0.8 * Z8_Sum) / \text{指しきい値}$

これらの値は、検出されたオブジェクトが正しい比率であることを保証します。

- Z8_sum 条件をチェックして、指の相対質量が指として認識されるのに十分な大きさであるかどうかを確認します。これは、隣接するセンサーで信号が検出されない場合に、セグメント内の非常に高いノイズを破棄するために行われます。
- Z9_sum 条件をチェックして、指の絶対質量が指として認識されるのに十分な大きさであるかどうかを確認します。Z8 状態と同様に、Z9 状態は、ノイズによる誤接触を防ぐことができます。

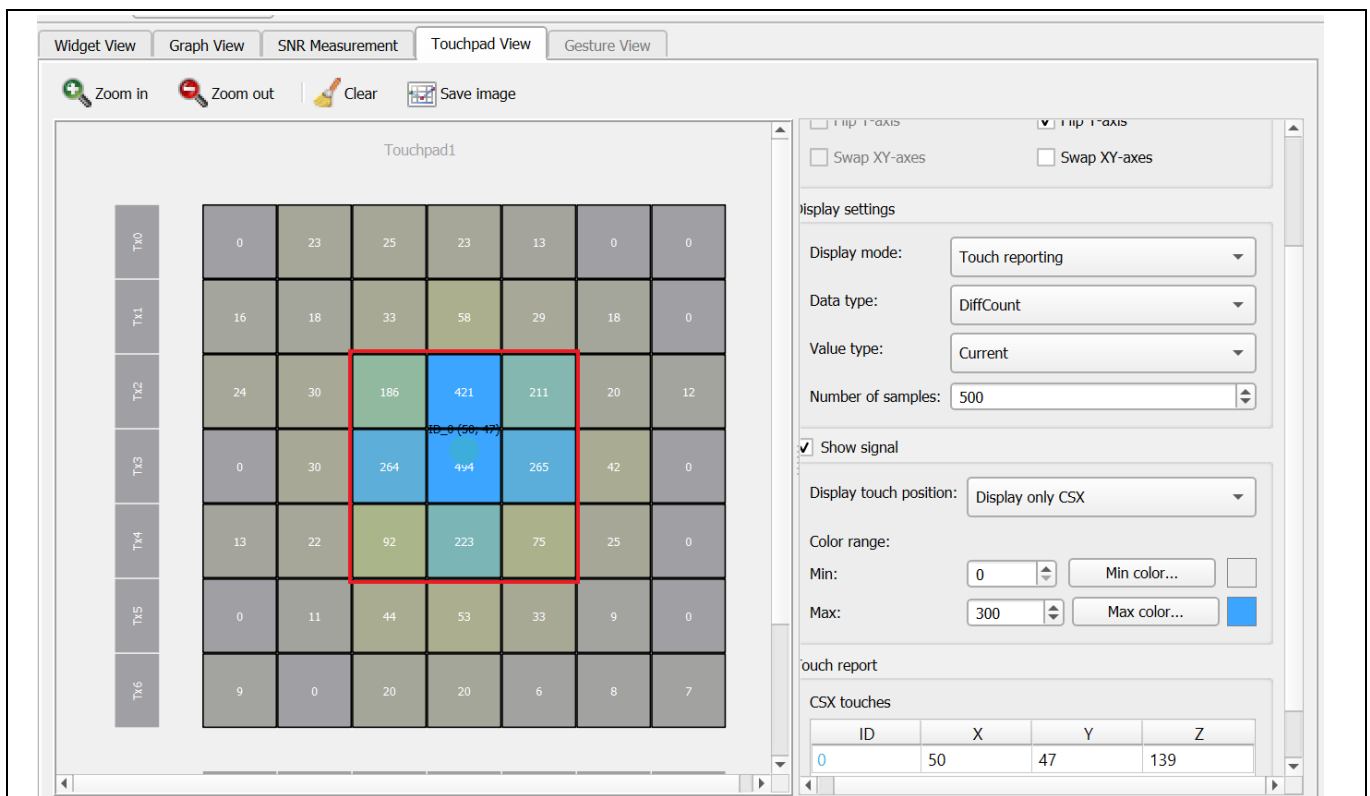


Figure 80 CAPSENSE™チューナーで得られた 3x3 マトリックス

CAPSENSE™性能のチューニング

Figure 81 に、CSX タッチパッドの調整手順の概要を示します。

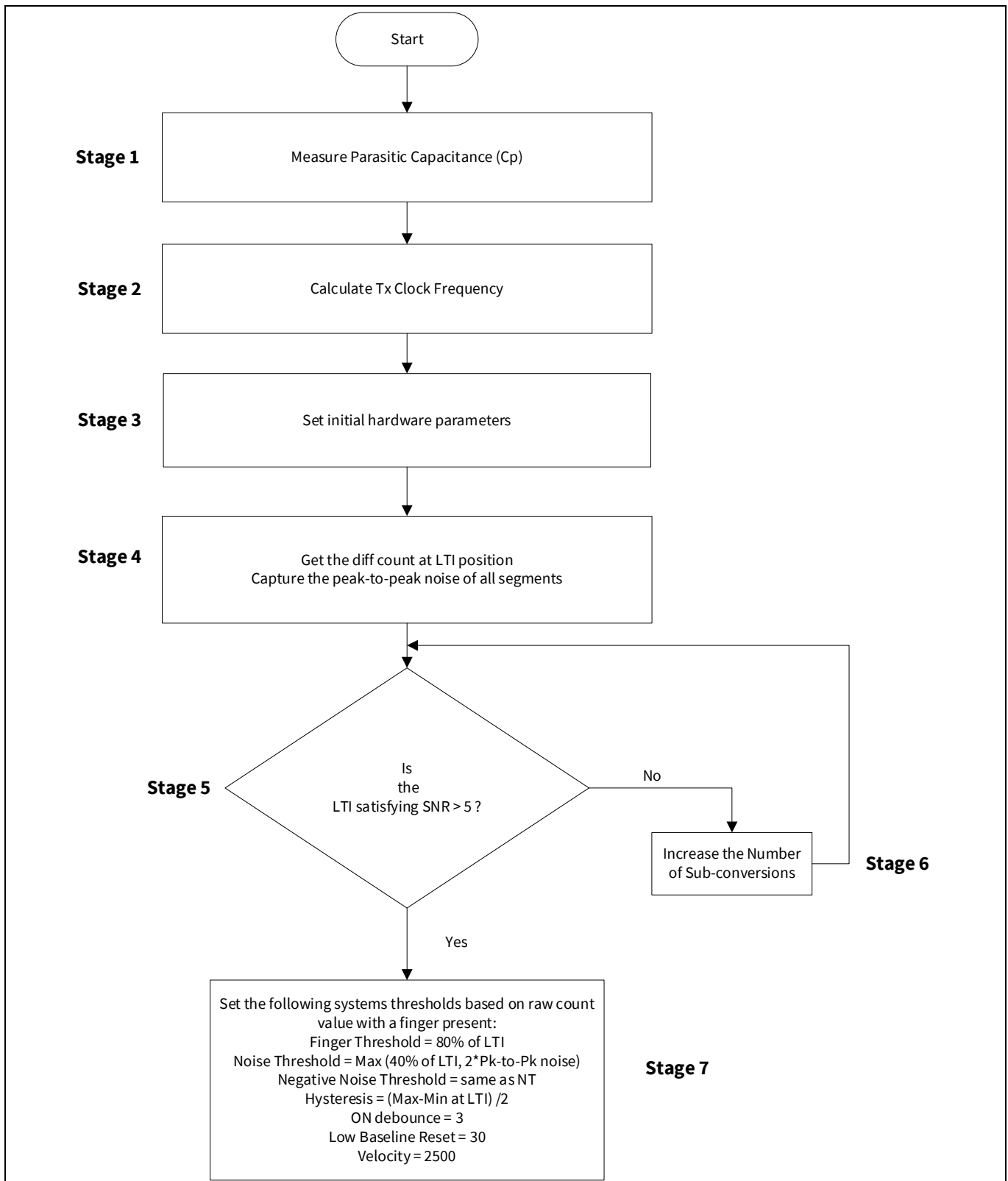


Figure 81 CSX タッチパッドウィジェットのチューニングフローチャート

LTI は、指で触れたときに 4 つのセンサーの中央にあるときのピーク差分カウントを測定します。LTI 信号数は、4 つのピークセンサーの平均です。これにより、有効性が最も低いタッチ信号が得られます。

CAPSENSE™性能のチューニング

5.3.4 CSD-RM センシング方式 (第 5 世代)

ここでは、CSD-RM センシング方式 (第 5 世代) を使用した手動チューニングの基本について説明します。また、CSD-RM センシング方式に影響を与えるハードウェアとソフトウェアのパラメーター、およびボタン、スライダー、タッチパッド、および近接ウィジェットの手動チューニングの手順についても説明します。

5.3.4.1 基本

5.3.4.1.1 変換利得と CAPSENSE™信号

変換ゲインは、センサーに指で触れたときにシステムが認識する信号の量に影響します。より多くのゲインがある場合、信号はより高くなり、より高い信号はより高い達成可能な**信号対ノイズ比 (SNR)**を意味します。ゲインを上げると、信号とノイズの両方が増える可能性があることに注意してください。ただし、必要に応じて、ファームウェアフィルターを使用してノイズを減らせます。使用可能なファームウェアフィルターの詳細については、**Table 7**を参照してください。

シングル CDAC での変換利得

シングル CDAC モードでは、raw カウントはセンサーの静電容量に正比例します。

式 47. センサー容量に対する raw カウントの関係

$$\text{raw count} = G_{\text{CSD}} C_s$$

ここで、

C_s = センサーの静電容量

$C_s = C_F$ (センサーに指がない場合)

$C_s = (C_P + C_F)$ (センサーに指がある場合)

G_{CSD} = CAPSENSE™ CSD の容量からデジタル変換利得

式 18 および **式 47** によるこの変換利得の概算値は、**式 48** を使用して示されます。

式 48. 静電容量からデジタル変換利得

$$G_{\text{CSD}} = \text{MaxCount} \frac{1}{\text{SnsClk}_{\text{Div}} \cdot C_{\text{ref}}}$$

ここで、 $\text{Maxcount} = N_{\text{Sub}} * \text{SnsClk}_{\text{Div}}$

式 48 および **式 47** による、シングル CDAC モードでの raw カウントの式を **式 49** に示します。

式 49. シングル CDAC モードの raw カウント

$$\text{raw count} = N_{\text{Sub}} \frac{C_s}{C_{\text{ref}}}$$

ここで、

N_{Sub} = サブ変換数

$\text{SnsClk}_{\text{Div}}$ = センスクロック分周器

C_s = センサーの静電容量

CAPSENSE™性能のチューニング

C_{ref} = 基準静電容量

$$C_{ref} = RefCDACCode * C_{lsb}$$

RefCDACCode = 基準 CDAC 値

$$C_{lsb} = 8.86fF$$

変換利得の調整可能なパラメータは、 C_{ref} , $SnsClk_{Div}$, および N_{Sub} です。Figure 82 に、raw カウントとセンサー容量のプロットを示します。

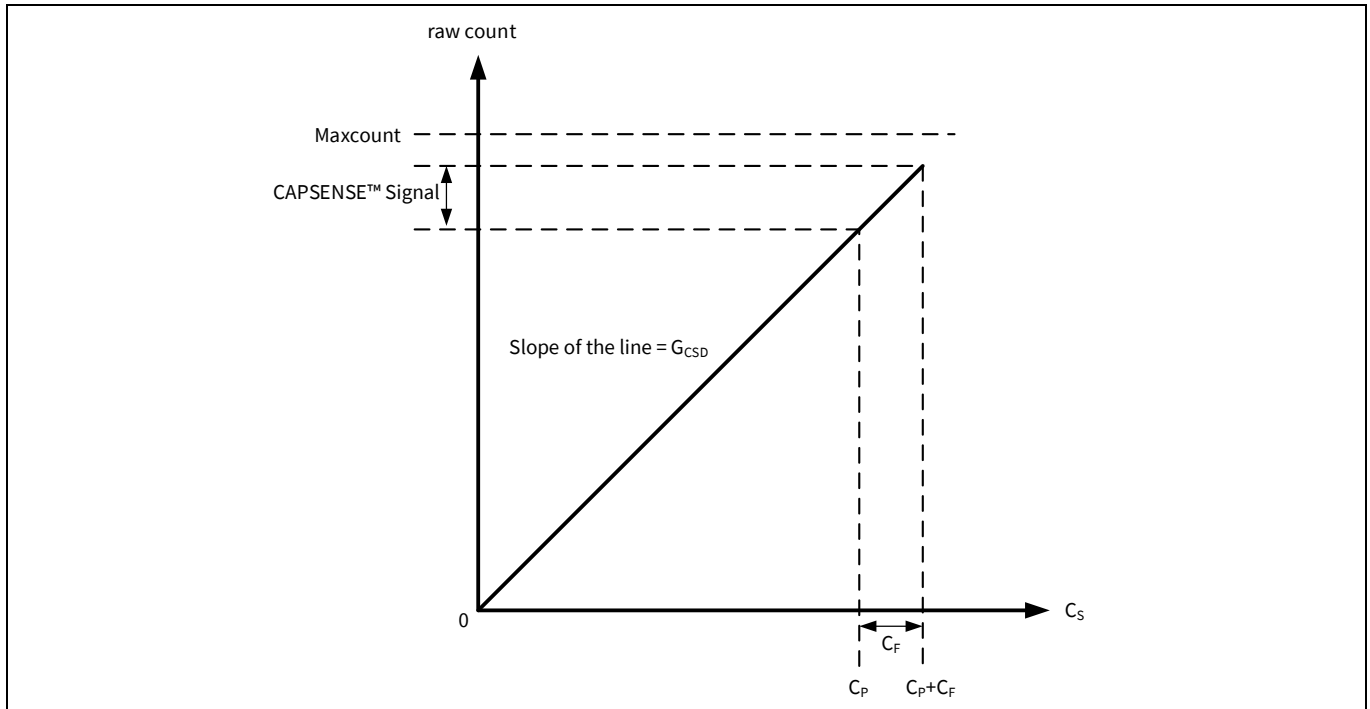


Figure 82 raw カウントとセンサー容量

指をセンサーに置いたときの raw カウントの変化は、CAPSENSE™信号と呼ばれます。Figure 83 に、変換利得に対して信号の値がどのように変化するかを示します。

CAPSENSE™性能のチューニング

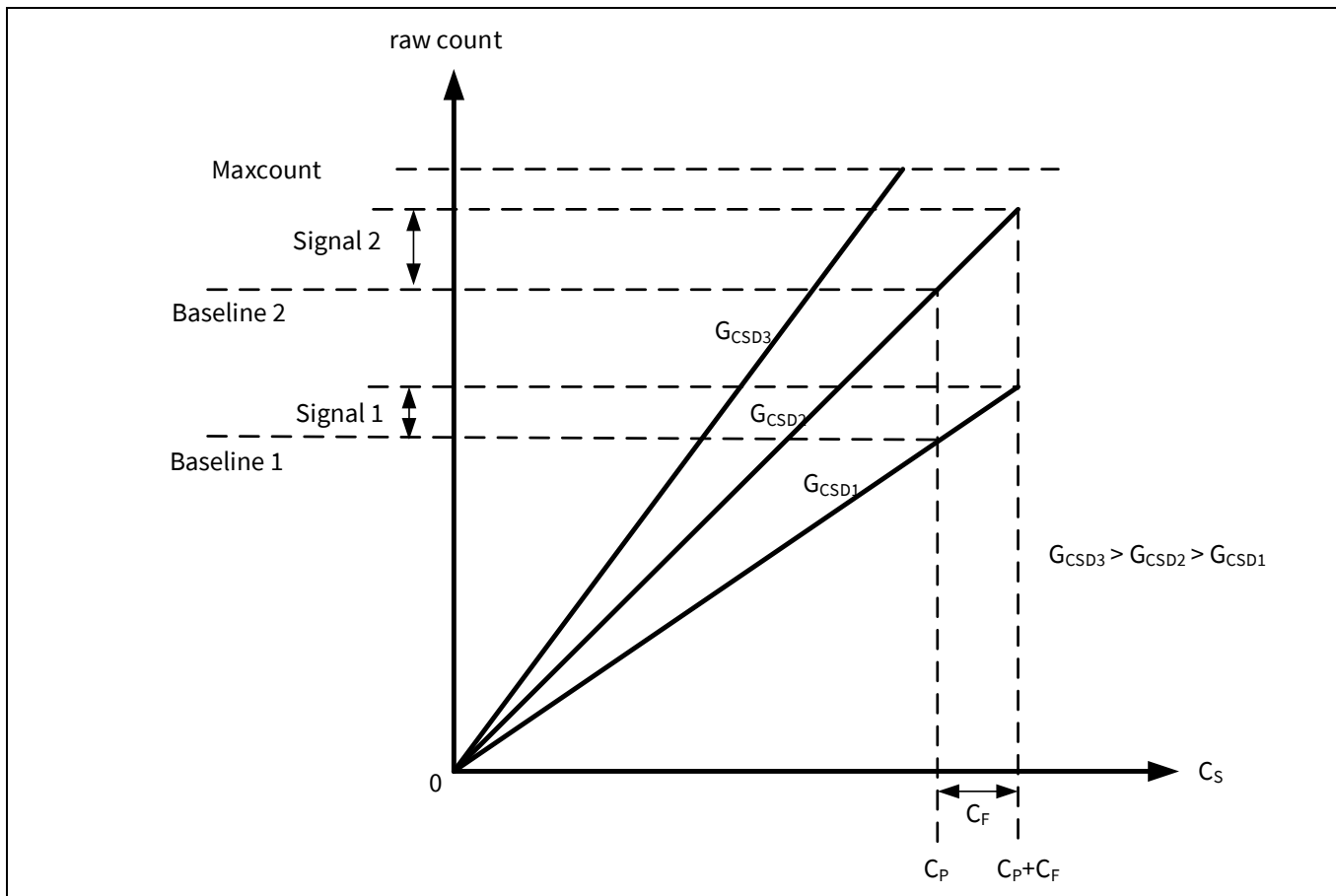


Figure 83 さまざまな変換利得の信号値

Figure 83 に、3つの変換ゲイン値 G_{CSD3} , G_{CSD2} , および G_{CSD1} に対応する3つのプロットを示します。変換利得を上げると、信号値が高くなります。ただし、この変換利得の増加により、 C_p に対応する raw カウント (つまり、ベースライン) も raw カウントの最大値 (Maxcount) に向かって移動します。利得値が非常に高い場合、 G_{CSD3} のプロットが示すように、raw カウントは飽和します。したがって、生のカウントの飽和を回避しながら、適切な信号値を取得するように変換利得を調整します。Figure 84 に示すように、センサーに指がない場合、つまり $C_s = C_p$ の場合、raw カウント = Maxcount の 85% になるように、CSD-RM パラメーターを調整します。これにより、最大の利得が保証され、環境の変化によって raw カウントが増加するのに十分なマージンがあり、指で触れても飽和しません。

CAPSENSE™性能のチューニング

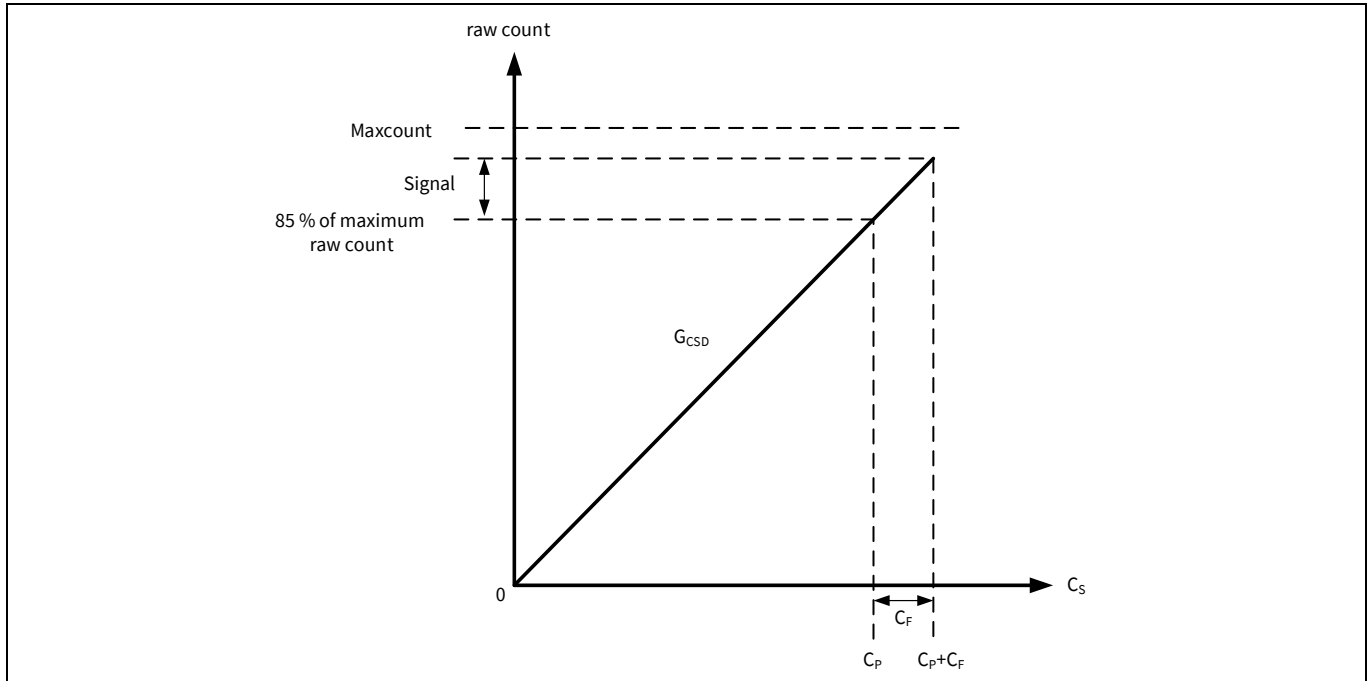


Figure 84 推奨チューニング

デュアル CDAC モードでの変換利得

式 19 および式 47 によるデュアル CDAC モードでの raw カウントの式を式 50 に示します。

式 50. デュアル CDAC モードでの raw カウント

$$\text{raw count} = G_{\text{CSD}} C_S - \text{Maxcount} * \frac{2 * C_{\text{comp}}}{C_{\text{ref}} \text{CompCLK}_{\text{div}}}$$

ここで、

$$\text{Maxcount} = N_{\text{Sub}} * \text{SnsClk}_{\text{Div}}$$

SnsClk_{Div} = センスクロック分周器

N_{Sub} = サブ変換数

C_{ref} = 基準静電容量

C_{comp} = 補償静電容量

CompCLK_{Div} = CDAC 補償分周器

C_S = センサー静電容量

$$C_{\text{ref}} = \text{RefCDACCode} * C_{\text{lsb}}$$

$$C_{\text{comp}} = \text{CompCDACCode} * C_{\text{lsb}}$$

RefCDACCode = 基準 CDAC 値

CompCDACCode = 補償 CDAC 値

$$C_{\text{lsb}} = 8.86\text{fF}$$

G_{CSD} は式 48 で与えられます。

CAPSENSE™性能のチューニング

シングル CDAC モードとデュアル CDAC モードの両方で、CSD-RM パラメーターを調整して、センサーに指がない場合、つまり $C_s = C_p$ の場合、Figure 85 に示すように、raw カウント = Maxcount の 85% になります。これは、フラットスポットを回避し、環境変化による raw カウントの飽和を回避する高い変換利得を確保します。

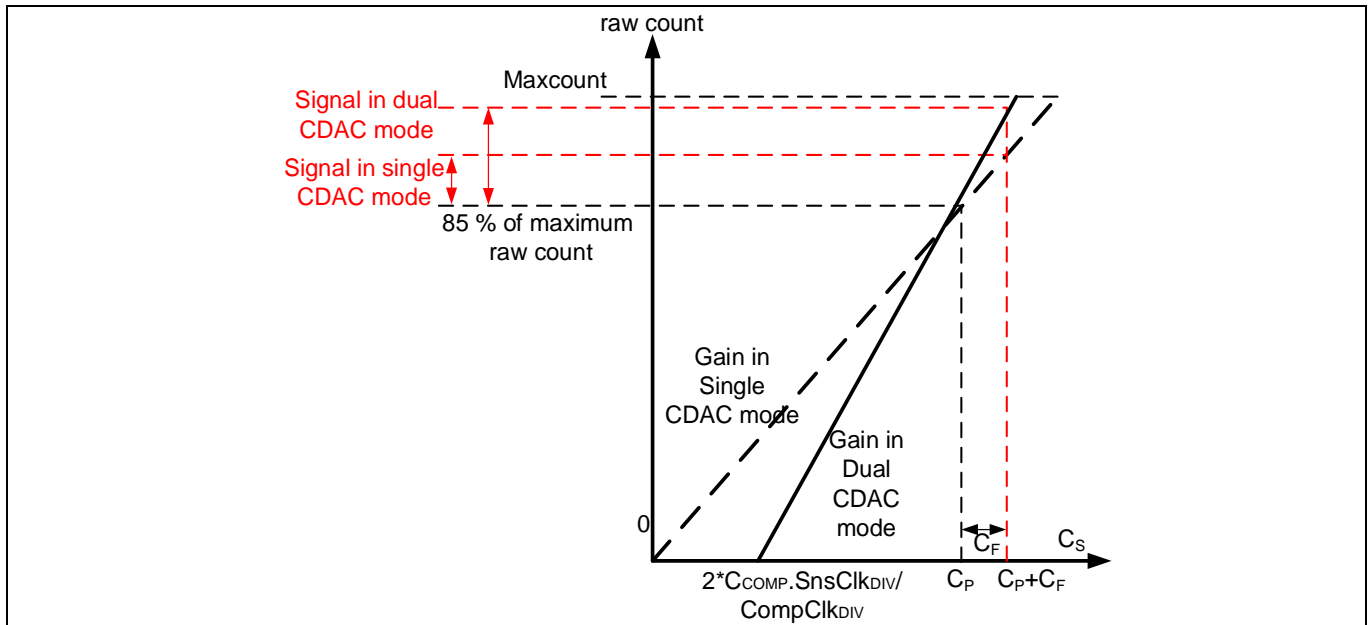


Figure 85 デュアル CDAC モードでの推奨チューニング

Figure 85 に示すように、85%の要件は、シングル CDAC モードでは固定利得に制限されますが、デュアル CDAC モードでは、 C_s 軸切片を右に移動し (CompClkDIV を増やすことにより)、それに応じて変調器を減らすことにより、利得を増やせます。CDAC スwitching (SnsClk_{Div}) は、 $C_s = C_p$ の場合でも raw カウント = Maxcount の 85% を達成します。このようにデュアル CDAC モードを使用すると、raw カウントと C_p のグラフに次の変更が加えられます。

a. 補償 CDAC を使用すると、式 51 に示すように、 C_s 軸にゼロ以外の切片が導入されます。

式 51. C_{comp} に関する C_s 軸切片

$$C_s \text{ axis intercept} = \left(\frac{2 * C_{comp} SnsClk_{Div}}{CompClk_{Div}} \right)$$

b. デュアル CDAC モードの C_{ref} の値は、シングル CDAC モードの C_{ref} の値の半分であるため (他のすべてのパラメーターは同じままです)、デュアル CDAC モードの利得 G_{CSD} は、式 25 によりシングル CDAC モードの利得の 2 倍になります。したがって、デュアル CDAC モードの信号は、特定の数のサブ変換 (N_{sub}) に対してシングル CDAC モードの信号の 2 倍になります。

センサーを手動で調整するときは、式 25, 式 27, および次の点を参照してください。

1. 利得が高いほど、感度が向上し、システム全体のパフォーマンスが向上します。ただし、利得 G_{CSD3} のプロットが Figure 59 に示すように、raw カウントが飽和するように利得を設定しないでください。シングル CDAC モードとデュアル CDAC モードの両方で、 C_p に対応する raw カウントが最大 raw カウントの 85% になるように利得を設定することを推奨します。
2. センスクロック周波数 (F_{sw}) は慎重に設定する必要があります。周波数が高いほど、 C_s サンプリングレートが高くなり、 F_{sw} 期間が長くなり、ノイズの平均化が向上しますが、式 31 に示すように、センサーを完全に充電および放電するには、周波数を十分に低くする必要があります。

CAPSENSE™性能のチューニング

3. 補正の有効化 CDAC (ベースライン補償) は、利得を上げる上で大きな役割を果たします。デュアル CDAC モードの変換利得で推奨されているように設定すると、利得が2倍になります。常に補正 CDAC を有効にし、補償 CDAC を有効にするときは、キャリブレーションされた C_{ref} が有効な範囲内にあることを確認してください。
4. リファレンス CDAC を低くし、ゲインを高くします。CDAC を調整して最高のゲインを達成しますが、Figure 60 および Figure 61 に示すように、 C_p に対応する raw カウントに、温度変化などの環境変化に対して十分なマージンがあることを確認してください。
5. スキャンに使用されるサブ変換数を増やすと、利得が増加します。サブ変換の数を増やすと、式 8 に従ってスキャン時間も長くなります。スキャン時間と利得のバランスは、サブ変換の数 (N_{sub}) を使用して達成する必要があります。

5.3.4.1.2 フラットスポット

理想的には、Figure 58 と Figure 61 に示すように、raw カウントはセンサーの静電容量と線形関係にある必要があります。ただし、実際には、RM コンバーターには、フラットスポットまたはデッドゾーンとも呼ばれる非感度ゾーンがあります。センサーの静電容量値の範囲に対して、RM コンバーターは Figure 86 に示すのと同じ raw カウント値を生成する場合があります。この範囲は、デッドゾーンまたはフラットスポットとして知られています。

式 52 に、さまざまな CAPSENSE™ハードウェアパラメーターに対するフラットスポットの関係を示します。

式 52. フラットスポット幅

$$\text{Flatspots Width} \propto \frac{C_s^2}{C_{MOD}} \cdot \frac{F_{SW}}{F_{MOD} \cdot \text{Bal}\%}$$

ここで、

C_s = センサーの静電容量

C_{MOD} = 変調器コンデンサ

F_{SW} = センスクロック周波数

F_{MOD} = 変調器クロック周波数

Bal% = Rawcount キャリブレーションパーセンテージ

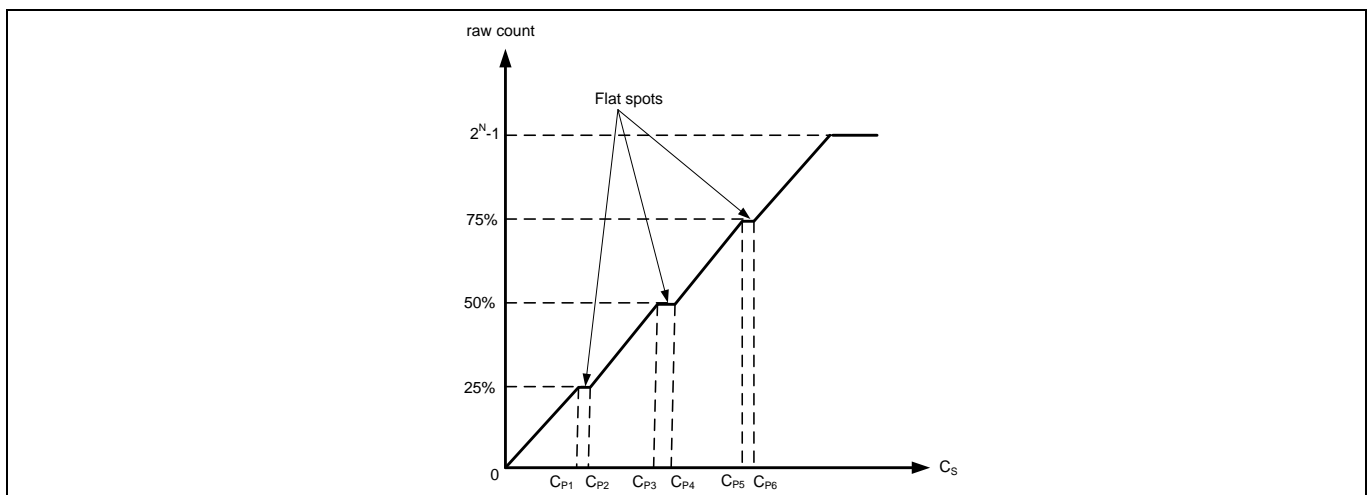


Figure 86 ダイレクトクロックを使用した場合の raw カウントとセンサー容量のフラットスポット

CAPSENSE™性能のチューニング

フラットスポット削減技術

1. rawcount キャリブレーションを 85%に設定します。

式 52 のように、フラットスポットの幅はキャリブレーションレベルに反比例します。キャリブレーションを 85%に設定すると、フラットスポットの幅が大幅に減少します。

2. デザリングを有効にします。

追加のデザ CDAC は、第 5 世代 CAPSENSE™アーキテクチャで利用できます。これにより、フラットスポット領域の周りで変換ポイントを移動するホワイトノイズが追加されます。

3. PRS クロックを有効にします。

これらのフラットスポットは、ダイレクトクロックがセンスクロックソースとして使用されている場合に顕著になります。PRS をセンスクロックソースとして使用すると、フラットスポットが減少します (SmartSense を使用したハードウェアパラメーターの決定セクションも参照してください)。PRS クロックは、より高い raw カウントキャリブレーションで信号または感度をわずかに低下させる可能性があります。PRS をクロックソースとして使用する場合は、raw カウントキャリブレーションを 65%に設定することを推奨します。

4. より大きな C_{MOD} を使用します。

フラットスポットの幅は、使用される C_{MOD} に反比例します。第 5 世代アーキテクチャは最大 10nF の C_{MOD} をサポートし、通常の値は 2.2nF です。また、 C_{MOD} を増やすと、ノイズ、初期化時間、および検出に必要な最小信号が増えるという悪影響があります。

5. センスクロック分周器を増やします。

センスクロック分周器を増やすと、フラットスポットの幅は狭くなりますが、スキャン時間は長くなります。フラットスポットが検出された場合は、スキャン時間の要件が満たされるようにセンスクロック分周器を増やします。

Table 13 に、推奨される優先順位およびその他の考慮事項におけるさまざまなフラットスポット削減手法を示します。

Table 13 フラットスポット削減技術

No.	フラットスポット削減技術	推奨	その他のメリット	デメリット
1	raw カウントキャリブレーションを 85%に設定	High	感度を向上させる	-
2	デザリングを有効	High	-	-
3	PRS クロックを有効	Low	EMI/EMC 放射と感受性を向上させる	Raw カウントキャリブレーションを 65%に設定する必要あり。感度が低下。
4	C_{MOD} を増加	Low	-	ノイズが増加
5	センスクロック分周器を増加	Low	-	スキャン時間が増加

5.3.4.2 CAPSENSE™ハードウェアパラメーターの選択

CAPSENSE™ハードウェアパラメーターは、変換利得と CAPSENSE™信号を制御します。Table 14 に、CSD-RM センシング方式に適用される CAPSENSE™ハードウェアパラメーターを示します。次の項では、

CAPSENSE™性能のチューニング

CAPSENSE™ CSD-RM システムの最適なパフォーマンスを実現するために、これらのパラメーターを調整する方法についてのガイダンスを提供します。

Table 14 CAPSENSE™コンポーネントのハードウェアパラメーター

No	ModusToolBox™での CAPSENSE™パラメーター
1	スキャンモード
2	Scan connection method
3	初期化サブ変換数
4	センスクロック分周器
5	センスクロックソース
6	変調器クロック分周器
7	基準 CDAC 値
8	CDAC 補償分周器
9	補償 CDAC 値
10	サブ変換数
11	Enable CDAC dither

セクション [5.3.4.2.1](#) および [5.3.4.2.2](#) は、ModusToolbox™の Eclipse IDE での CAPSENSE™パラメーターの選択を示します。CAPSENSE™の設定の詳細については、[コンポーネント データシート/ミドルウェア 文書](#)を参照してください。

5.3.4.2.1 SmartSense を使用したハードウェアパラメーターの決定

Table 14 に、CAPSENSE™ハードウェアパラメーターを示します。これらのパラメーターを手動で最適な値に調整することは、時間のかかる作業です。SmartSense を使用して、これらのハードウェアパラメーターを決定し、手動チューニングの初期値として使用できます。これらの値を微調整して、スキャン時間、SNR、消費電力をさらに最適化するか、CAPSENSE™システムの EMI/EMC 機能を向上させることができます。チューニングモードを SmartSense に設定し、指容量、[センスクロックソース](#)、[CDAC ディザ](#)以外のパラメーターの初期値を設定します。アプリケーションの要件に従ってこれらを設定します。

チューニング手順については [SmartSense](#) のセクションを参照し、チューナーGUI を使用して SmartSense によって設定されたすべてのハードウェアパラメーターを読み戻します。チューナーGUI の使用方法の詳細については、[CAPSENSE™チューナーガイド](#)を参照してください。

Figure 87 に、0.1pF の最小指容量を検出するために特定のハードウェア用に SmartSense によって調整されたチューナーGUI の最適なハードウェアパラメーター値を示します。

CAPSENSE™性能のチューニング

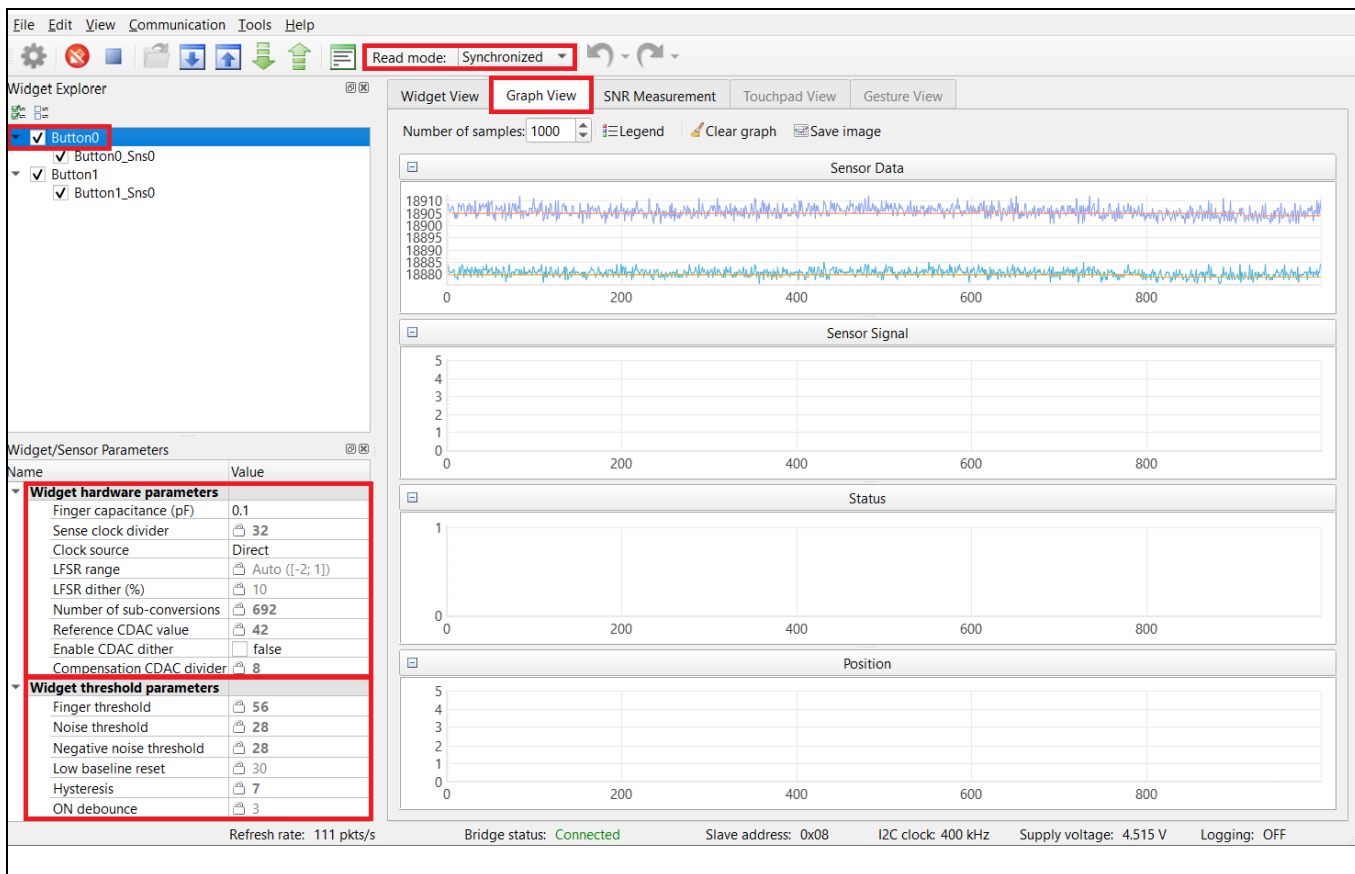


Figure 87 チューナーGUIでのハードウェアパラメーター値の読み戻し

5.3.4.2.2 ハードウェアパラメーターの手動調整

スキャンモード

スキャンモードは、CS-DMA または 割込み駆動モードとして設定できます。自律スキャンの場合は DMA モードを選択し、レガシー割込みベースのスキャンの場合は割込み駆動モードを選択します。

センサー接続方法

自動スキャンは CTRLMUX 方式でのみ使用できますが、この方式ではサポートされるピンの数が制限されます (対応するピンについては [デバイス データシート](#) を参照してください)。さらに、オンチップ IO ノイズに対する耐性が向上します。割込み駆動モードでより多くのピンをサポートするには、AMUXBUS 方式を選択してください。

変調器クロック周波数

変調器クロックは、「センサースキャン時間」とも呼ばれる静電容量からデジタルへの変換の変換時間を制御します (式 8 を参照)。

変調器のクロック周波数が低いということは、次のことを意味します。

- 変換時間が長くなります (式 55 および 式 57 を参照)。
- レシオメトリックコンバーターの積分時間が長いため、raw カウントのピークツーピークノイズが低くなる。
- より広いフラットスポット

CAPSENSE™性能のチューニング

ほとんどの場合、変換時間が最短になるように最高周波数を選択し、フラットスポットを狭くします。必要に応じて、低速の変調器クロックを使用して、raw カウントのピークツーピークノイズを低減します。

必要な変調器クロック周波数 (F_{MOD}) に基づいて、式 53 を使用して変調器クロック分周器を計算します。

式 53. 変調器クロック分周器

$$\text{変調器クロック分周器} = \frac{F_{\text{Clock}}}{F_{\text{MOD}}}$$

ここで、

F_{Clock} = CAPSENSE™ ブロックに接続されたクロック周波数

初期化サブ変換

初期化の一環として、 C_{MOD1} と C_{MOD2} を必要な電圧 ($V_{DDA}/2$) で充電する必要があります。初期化には、 C_{MOD} 初期化、 C_{MOD} ショート、および初期化サブ変換の 3 つのフェーズがあります。 C_{MOD} の初期化フェーズでは、 C_{MOD1} が GND にプルされ、 C_{MOD2} が V_{DDA} にプルされます。 C_{MOD} の短絡フェーズでは、両方のコンデンサが相互に接続されるため、電荷が共有されて、両方で $V_{DDA}/2$ に近い電圧が生成されます。2 つのフェーズの後、スキャンが開始されますが、初期化サブ変換数のために raw カウントは破棄されます。

初期化サブ変換数は、式 54 に基づいて選択する必要があります。

式 54. 初期化サブ変換数

$$\text{初期化サブ変換数} = \text{ceiling} \left(\frac{C_{\text{MOD}} * V_{\text{OS}}}{V_{\text{DDA}} * C_{\text{S}} * (1 - \text{Base}\%) * \left(\frac{1}{\text{Bal}\%} - 1 \right)} \right) + 1$$

または

$$\text{初期化サブ変換数} = \text{ceiling} \left(\frac{C_{\text{MOD}} * V_{\text{OS}}}{V_{\text{DDA}} * \text{SnsClkDiv} * C_{\text{ref}} * (1 - \text{Bal}\%)} \right) + 1$$

ここで、

C_{MOD} = 変調器コンデンサ

V_{OS} = コンパレータオフセット電圧 (3mV)

C_{S} = センサー静電容量

Base% = ベースライン補償率

Bal% = Raw カウントキャリブレーションパーセンテージ

$\text{SnsClk}_{\text{Div}}$ = センスクロック分周器

C_{ref} = 基準静電容量

$C_{\text{ref}} = \text{RefCDACCode} * C_{\text{lsb}}$

RefCDACCode = 基準 CDAC 値

$C_{\text{lsb}} = 8.86\text{fF}$

CAPSENSE™性能のチューニング

センスクロックパラメーター

センスクロックに関連する 2 つのパラメーターがあります。センスクロックソースとセンスクロック分周器です。

センスクロックソース

「自動」を選択すると、コンポーネントは、ウィジェットごとに Direct, PRSx, および SSCx から最適な Sense クロックソースを自動的に選択します。自動を選択しない場合は、以下に基づいてクロックソースを選択します。

- 特定の周波数での EMI/EMC ノイズを低減するには、SSCx (スペクトラム拡散クロック) モードを使用します。この機能は、PSoC™ 4S シリーズ, PSoC™ 4100S Plus, PSoC™ 4100PS, PSoC™ 4100S Max, および PSoC™ 6 ファミリのデバイスで利用できます。この場合、センスクロックの周波数は所定の範囲に広がります。
- 絶対容量の測定にはダイレクトクロックを使用してください。
- PRSx (疑似ランダムシーケンス) モードを使用して、フラットスポットを削除し、EMI/EMC 放射と感受性を改善します。第 5 世代 CAPSENSE™ では、PRS クロックにより、より高い raw カウントキャリブレーションパーセントで信号/感度がわずかに低下するため、PRS クロックを使用する場合は 65% raw カウントキャリブレーションを推奨します。

SSCx を選択するときは、SSCx クロックソース選択の [ModusToolbox™ CAPSENSE™ configurator guide](#) に記載されている条件が満たされるように、センスクロック周波数, 変調器クロック周波数, およびサブ変換数を選択する必要があります。

センスクロック分周器

Figure 46 に示すように、センサーが各センスクロック周期で完全に充電および放電するように、センスクロック分周器を選択する必要があります。第 5 世代の CSD-RM の場合、充電と放電は 1 クロック周期で 2 回行われることに注意してください。

これには、[式 55](#) に従って最大センスクロック周波数を選択する必要があります。

式 55. センスクロック最大周波数

$$F_S(\text{maximum}) = \frac{1}{4 * 5 * R_{\text{SeriesTotal}} C_p}$$

式 56. 総並列抵抗

$$R_{\text{SeriesTotal}} = R_{\text{EXT}} + R_{\text{internal}}$$

ここで、

C_p = センサーの寄生容量

$R_{\text{SeriesTotal}}$ = 内部スイッチの内部抵抗 R_{internal} を含む総並列抵抗、推奨される外部直列抵抗 560Ω (センサーパッドをデバイスピンに接続する PCB 配線に接続)、および高抵抗材料 (例 ITO または導電性インク)。

R_{internal} = 内部抵抗、これはスキャンモードとシールドモードによって異なります。[Table 15](#) を参照してください。

CAPSENSE™性能のチューニング

Table 15 センサーの内部抵抗

スキャンモード	R _{internal}
CTRLMUX	525 Ω
AMUXBUS	425 Ω

C_pの値は、CSDビルトインセルフテストAPIを使用して見積もることができます。詳細については、[コンポーネントデータシート/ミドルウェア文書](#)を参照してください。

式 57 に示すように、スキャン時間を最小化するには、式 55 の基準を満たす最大センスクロック周波数 (F_{sw}) を使用することをお勧めします。

式 57. センサースキャン時間

$$T_{\text{SCAN}} = \frac{N_{\text{Sub}}}{F_{\text{SW}}}$$

ここで、

N_{sub} = サブ変換数

必要なセンスクロック周波数 (F_{sw}) に基づいて、式 58 に従ってセンスクロック分周器を選択します。

式 58. センスクロック分周器

$$\text{センスクロック分周器} = \frac{F_{\text{Mod}}}{F_{\text{sw}}}$$

式 48 は、良好なゲインを得るには最大クロック周波数を使用するのが最適であることを示しています。ただし、Figure 46 に示すように、センサーコンデンサが完全に充電および放電することを確認する必要があります。また、クロック周波数が高くなると、充電と放電が増えるため、消費電流が増加することに注意してください。

一般的に、シールド電極の C_p はセンサー C_p と比較して高くなります。良好な耐液性を得るには、シールド信号がシールド電極の調整セクションに記載されている条件を満たす必要があります。満たされない場合は、センスクロック周波数をさらに下げて条件を満たします。

サブ変換数

サブ変換数によって、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。固定変調器クロックとセンスクロックの式 19 から、サブ変換 (N_{sub}) の数を増やすと、信号と SNR が増加します。ただし、サブ変換数を増やすと、式 59 に従ってセンサーのスキャン時間も長くなります。

式 59. CSD-RM スキャン時間

$$\text{Scan time} = \frac{N_{\text{Sub}}}{F_{\text{SW}}}$$

最初に、値を低い数値に設定し、チューナーGUIを使用してセンサーの SNR を見つけます。選択した N_{sub} で SNR が 5:1 を超えていない場合は、SNR 要件が満たされるように段階的に N_{sub} を増やしてみてください。

容量性 DAC

第 5 世代 CAPSENSE™ は、Figure 44 に示すように、C_{MOD} のバランスをとる基準 CDAC (C_{ref}) と補償 CDAC (C_{comp}) の 2 つの CDAC をサポートします。これらは、容量からデジタルへの変換のための変換利得と

CAPSENSE™性能のチューニング

CAPSENSE™信号変換利得と CAPSENSE™を制御します。CAPSENSE™コンポーネントを使用すると、CDACの次の設定が可能になります。

- 補償 CDAC の有効化または無効化
- CDAC の自動キャリブレーションの有効化または無効化
- 補償 CDAC 分周器、自動キャリブレーションが無効になっている場合の基準および補償 CDAC の DAC コード選択

基準 CDAC (C_{ref})

基準 CDAC は、 C_{MOD} からセンサーの自己容量 (C_S) によって転送された電荷を補償するために使用されます。切り替え回数は、センサーの自己容量によって異なります。センサーの上に指を置いた場合、補正するために追加の基準 CDAC スwitchingが必要です。

C_{ref} は以下の基準を満たす必要があります。

- 補償が無効の場合:
 $RefCDACCode \geq 25$
- 補償が有効な場合:

$$RefCDACCode \geq \frac{20}{CDAC \text{ Compensation Divider}}$$

ここで、

$C_{ref} = \text{基準静電容量} = RefCDACCode * Clsb$

$RefCDACCode = \text{基準 CDAC 値}$

$Clsb = 8.86fF$

補償 CDAC (C_{comp})

補償 CDAC を有効にすることは「デュアル CDAC」モードと呼ばれ、**変換利得と CAPSENSE™信号**で説明されているように信号が増加します。ほとんどの場合、補償 CDAC を有効にします。

補償コンデンサは、センサーからの過剰な自己容量を補償して感度を上げるために使用されます。切り替えられる回数は、ユーザーアプリケーションがセンサーの自己容量から補償 (除去) しようとしている電荷の量によって異なります。

補償 CDAC 分周器

シングルセンスクロックで補償コンデンサが切り替わる回数を K_{comp} で表します。以下の基準が満たされるように、**式 60** に基づいて CDAC 補償分周器を選択します。

式 60. CDAC 補償分周器

$$CDAC \text{ 補償分周器} = \frac{\text{Sense clock divider}}{K_{comp}}$$

自動キャリブレーション

自動キャリブレーション機能により、ファームウェアは CDAC を自動的にキャリブレーションして、必要なキャリブレーション目標である 85% を達成できます。ほとんどの場合、自動キャリブレーションを有効にすることを推奨します。この機能を有効にすると、次のようになります。

- パーツ間の C_p の変動があっても、raw カウントのキャリブレーションを最大 raw カウントの 85% に修正
- **フラットスポット**の影響を減らす

CAPSENSE™性能のチューニング

設計環境に大きな温度変化が含まれる場合、85%CDAC キャリブレーションレベルが高すぎ、温度の大きな変化に対して raw カウントが飽和しやすくなり、SNR が低下することがあります。この場合、ファームウェアで `Cy_CapSense_CalibrateAllSlots()` を使用して、キャリブレーションレベルを低く調整してください。

さまざまな環境条件下で CAPSENSE™ を適切に機能させるために、非常に低いまたは高い CDAC コードを回避することを推奨します。8 ビット CDAC の場合、可能な 0~255 の範囲の 6~200 の CDAC コードを使用することを推奨します。CAPSENSE™ チューナーを使用して、自動キャリブレーションされた CDAC 値がこの推奨範囲内にあることを確認できます。CDAC 値が **式 47**、**式 48**、および **式 50** に基づく推奨範囲外の場合は、キャリブレーションレベルまたは F_{mod} または F_{sw} を変更して、CDAC コードを適切な範囲に収めることができます。

リセット時に raw カウントレベルを測定して C_p の変化を検出する必要がある場合は、CDAC 自動キャリブレーションを無効にします。以下は例です。

- ボード間またはレイアウトの問題によるセンサー C_p の大きな変動の検出
- リセット時の指の接触の検出
- 液面検知などの高度な CAPSENSE™ メソッド。例えば、リセット時に液面レベルごとに raw カウントレベルを変えられます。

CDAC コードの選択

これは推奨されるアプローチではありません。ただし、このアプローチは、何らかの理由で自動キャリブレーションを無効にする場合にのみ使用できます。CDAC コードを取得するには、最初に自動キャリブレーションを有効にして CAPSENSE™ コンポーネントを構成し、他のすべてのハードウェアパラメータを最終調整に必要なものと同じにして、**チューナー GUI** を使用してキャリブレーションされた CDAC 値を読み戻します。次に、CAPSENSE™ コンポーネントを再設定して自動キャリブレーションを無効にし、取得した CDAC コードを **チューナー GUI** から読み戻される固定 DAC コードとして使用します。

CDAC ディザ

入力容量がスイープされると、raw カウントは容量とともに直線的に増加するはずですが、raw カウントが入力容量によって線形に変化しない領域があります。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、ディザリング 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、フラット領域の周りで変換ポイントを移動するホワイトノイズを追加します。

5.3.4.2.3 シールド電極の調整

液体耐性またはセンサーの C_p を下げるために、CSD-RM センシング方式でシールド電極を有効にする場合は、シールド関連のパラメータを追加で設定または調整する必要があります。

シールド電極調整理論

理想的には、シールド波形は、**CAPSENSE™ CSD-RM シールド** で説明されているセンサーの波形と正確に同じである必要があります。ただし、実際のアプリケーションでは、シールド波形の整定時間が長くなる場合があります。オシロスコープでセンサーとシールドの波形を観察します。波形の例を **Figure 88** と **Figure 89** に示します。シールド波形は、センスクロック波形のオン時間の 90% 以内にセンサー電圧が落ち着く必要があり、VREF に対するシールド信号のオーバーシュートエラーは 10% 未満である必要があります。

これらの条件が満たされない場合、シールドハッチに触れたときにセンサーの raw カウントの変化が観察されます。さらに、**非アクティブなセンサー接続** で説明したように非アクティブセンサーがシールドに接続されている場合、1 つのセンサーに触れると、他のセンサーの raw カウントが変化する可能性が

CAPSENSE™性能のチューニング

あります。これは、シールド電極が適切に調整されていない場合にクロストークが発生することを示します。

シールド波形の正しい充電と放電を保証するおおよその最大シールド周波数 (F_{Shield}) は、**式 61** を使用して計算できます。

式 61. センスクロック最大周波数

$$F_{\text{Shield}}(\text{maximum}) = \frac{1}{4 * 5 * R_{\text{SeriesTotal}} C_{\text{sh}}}$$

ここで、

C_{sh} = シールド C_p

$R_{\text{SeriesTotal}} = R_{\text{internal}} + R_{\text{EXT}}$

R_{EXT} = シールド電極に接続された外部直列抵抗。推奨値は 560Ω です。

R_{internal} = 内部抵抗。これはスキャンモードとシールドモードによって異なります。**Table 16** を参照してください。

Table 16 シールドセンサーの内部抵抗

スキャンモード	R_{internal} (アクティブシールド)	R_{internal} (パッシブシールド)
CTRLMUX	250 Ω	250 Ω
AMUXBUS	300 Ω	100 Ω

SmartSense では、センスクロック周波数が自動的に設定されます。これらの条件が満たされているかどうかを確認してください。満たされない場合は、**手動チューニング**に切り替え、これらの条件が満たされるようにセンスクロック周波数を手動で設定してください。

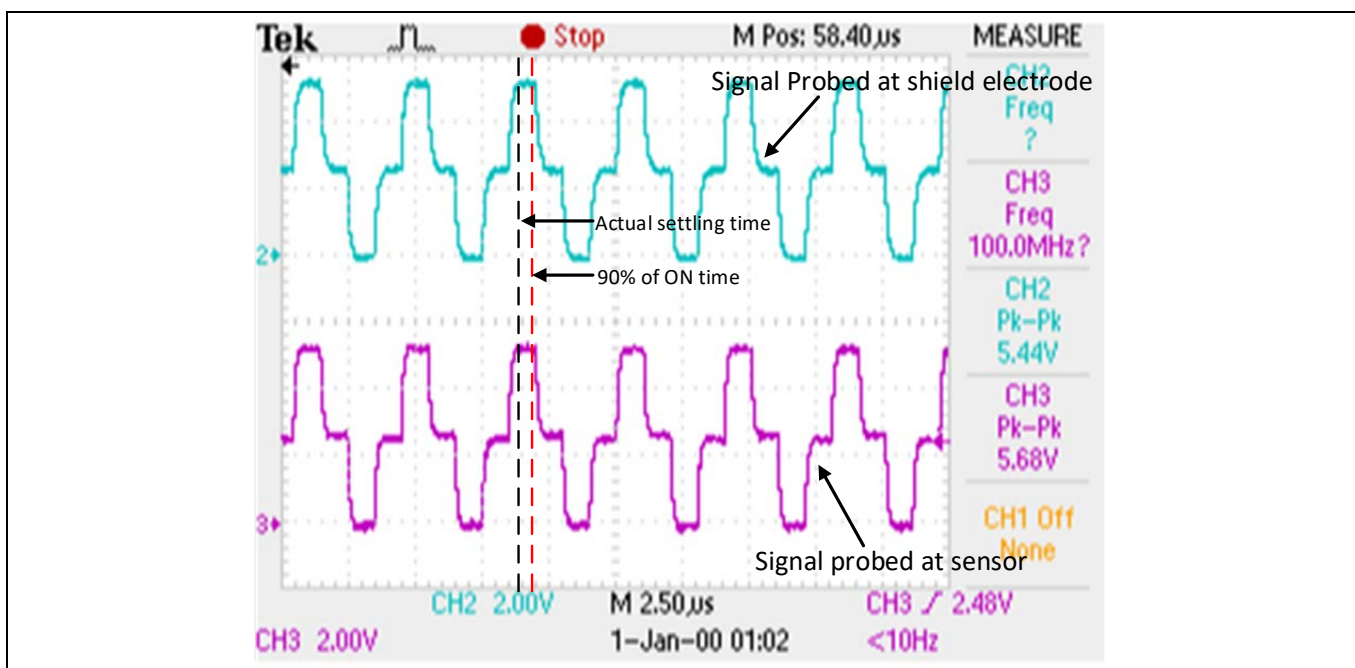


Figure 88 適切に調整されたシールド波形 (アクティブシールド)

CAPSENSE™性能のチューニング

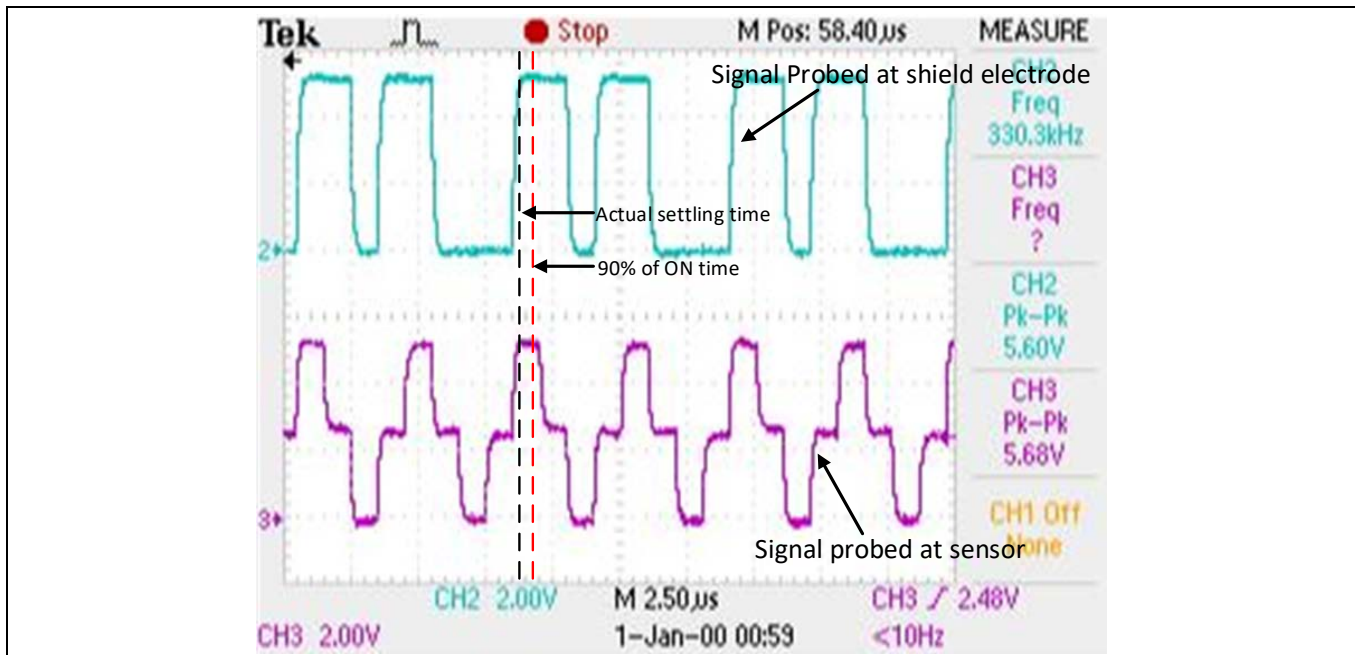


Figure 89 適切に調整されたシールド波形 (パッシブシールド)

シールド関連のパラメーターの調整

非アクティブなセンサー接続

シールド電極が耐液性設計で有効になっている場合、またはシールドを使用してセンサーの寄生容量を低減する場合は、このオプションを「Shield」として指定する必要があります。それ以外の場合は、「Ground」を選択します。

ただし、非アクティブなセンサーがシールドに接続されるため、放射エミッションが高くなるリスクがあります。このような状況では、CAPSENSE™APIを使用して、非アクティブなセンサー接続を手動で制御します。未使用のすべてのセンサーをシールドに接続する代わりに、放射状の放射を減らすために、スキャン対象のセンサーの近くにある非アクティブなセンサーまたは非アクティブなセンサーのみをシールドに接続します。

シールド電極数 (シールド総数)

このパラメーターは、設計に必要なシールド電極の数を指定します。ほとんどの設計は、1つの専用シールド電極で機能します。ただし、一部の設計では、PCBレイアウトのルーティングを容易にするため、またはシールド層に使用されるPCBの領域を最小限に抑えるために、複数の専用シールド電極が必要です。シールド電極のレイアウトガイドラインを参照してください。 [シールド電極のレイアウトガイドライン](#)。

シールドモード

第5世代CAPSENSE™アーキテクチャは、アクティブシールドとパッシブシールドの2つのシールドモードをサポートしています。アプリケーションに最適なモードを決定するには、[CAPSENSE™ CSD-RM シールド](#)のセクションを参照してください。

5.3.4.3 CAPSENSE™ソフトウェアパラメーターの選択

第5世代のCAPSENSE™ソフトウェアパラメーターは、第4世代のパラメーターと同じです。したがって、これらのパラメーターは、[CAPSENSE™ソフトウェアパラメーターの選択](#)のセクションで説明したように選択できます。

CAPSENSE™性能のチューニング

5.3.4.4 自律スキンの設定

自律スキンは、センサースキャン間の CPU 介入の必要性を排除することにより、CPU オフロードを改善します。Figure 90 に、すべてのスロットをスキャンする波形を示します。これは、自律スキンの要件を備えた CAPSENSE™CPU 帯域幅を示します。自律スキンの場合、CPU がすべてのスロットのスキャンコマンドを開始すると、すべてのスロットのスキャンが完了するまで、CAPSENSE™によって CPU 割り込みが発生することはありません。ただし、割り込み駆動型スキャンでは、各スロットスキャンの後に、次のスロットセンサーを構成するために CPU 割り込みが発生します。

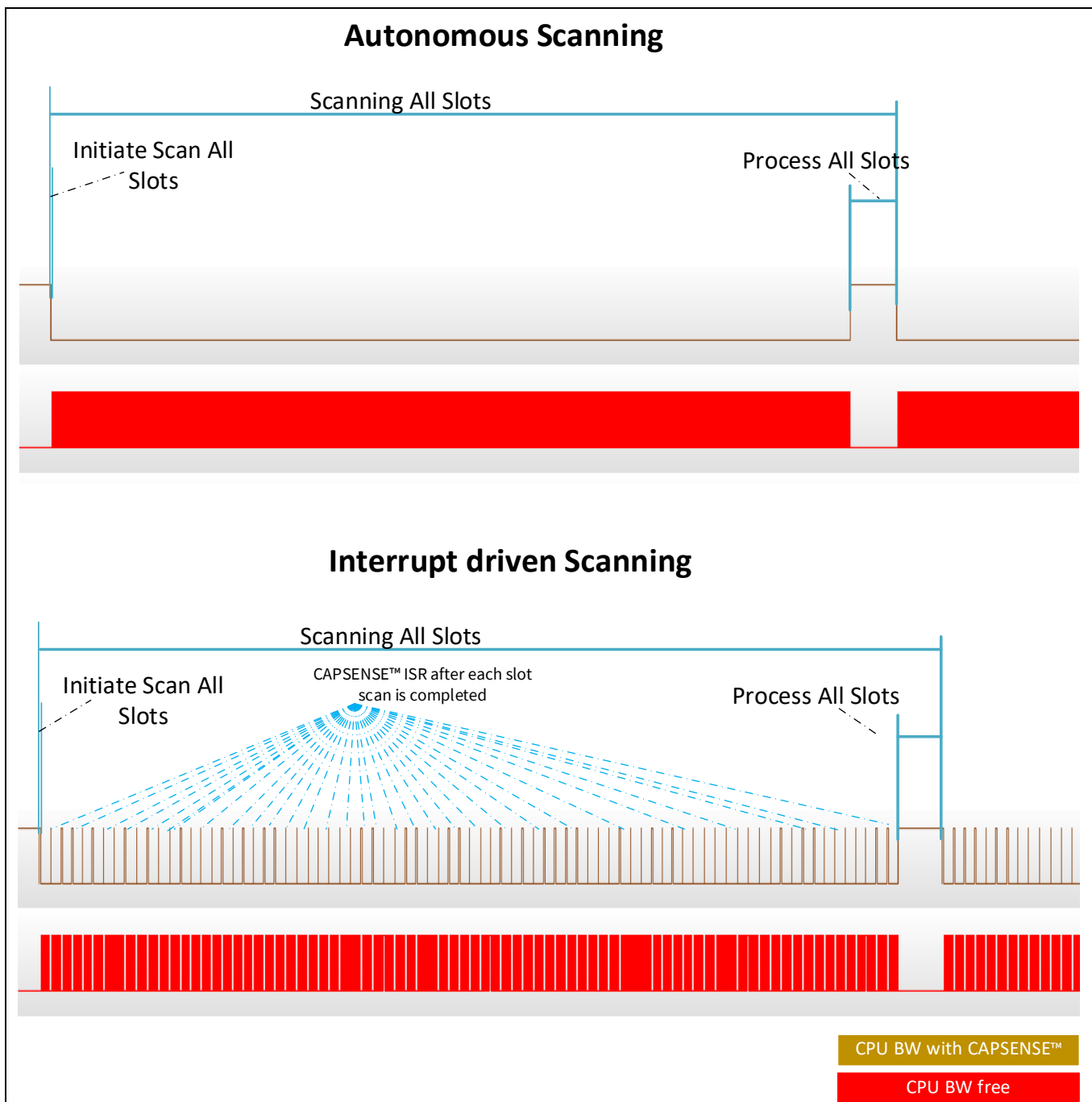


Figure 90 スキャンモードの CAPSENSE™コンフィギュレーター設定

自律スキンは、Figure 91 に示すように、CAPSENSE™コンフィギュレーターでスキャンモードが連鎖スキャン-DMA (CS-DMA) として設定されている場合にのみ使用できます。また、センサー接続方法は

CAPSENSE™性能のチューニング

CTRLMUX としてのみ使用できます。これにより、使用可能な CAPSENSE™センサーの数が制限されます。割込み駆動モードでは、センサー接続は AMUXBUS または CTRLMUX のいずれかとして設定できます。AMUXBUS を介して、任意の GPIO ピンを CAPSENSE™センサーとして構成できますが、次のすべてのセンサーを構成してスキャン結果を読み出すには、CPU 割込みを処理する必要があります。

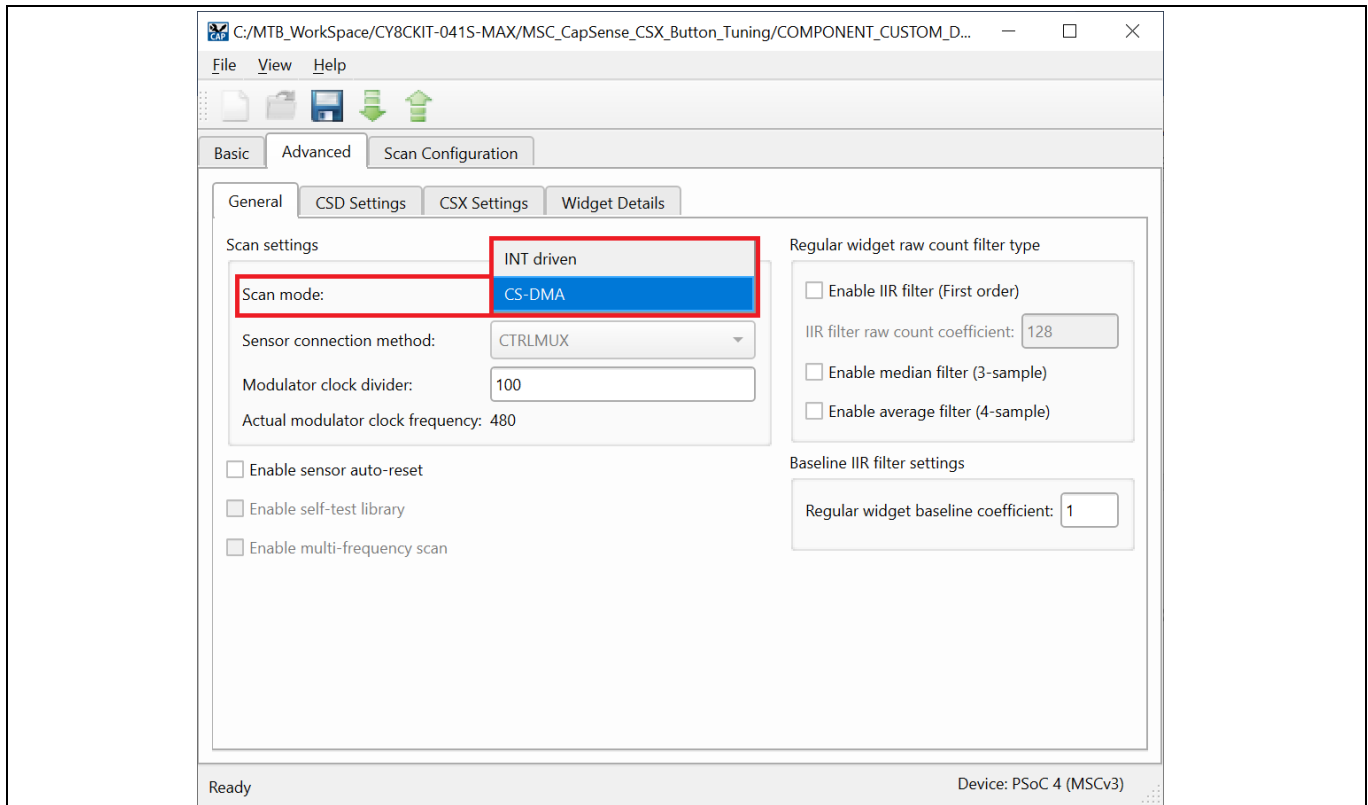


Figure 91 スキャンモードの CAPSENSE™コンフィギュレーター設定

5.3.4.4.1 連鎖スキャン-DMA

連鎖スキャン-DMA モードでは、DMA が各センサーの構成を処理するため、各センサースキャンの完了後に CPU の介入が不要になります。Figure 92 に示すように、MSC ブロックの各チャンネルでは、デバイスコンフィギュレーターで 4 つの DMA チャンネルを設定する必要があります。

CAPSENSE™性能のチューニング

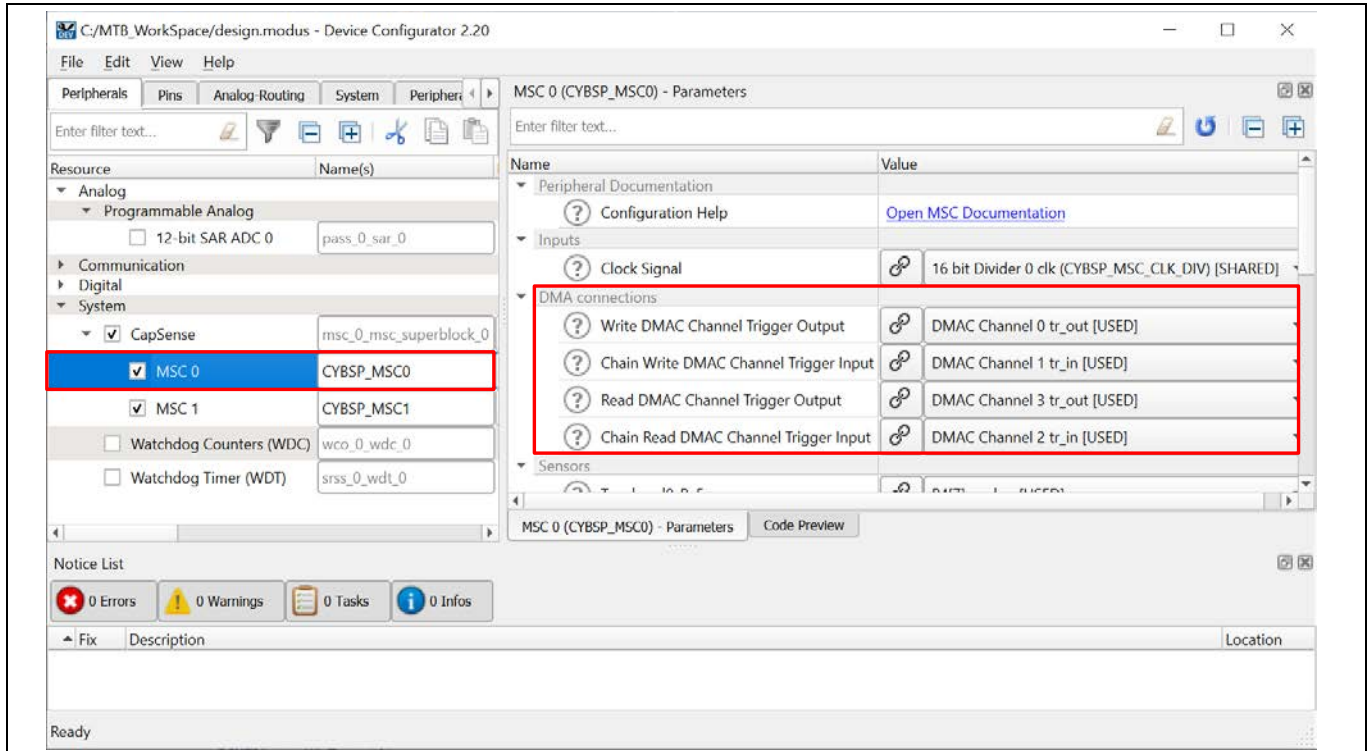


Figure 92 MSC チャンネルの DMA 構成

Figure 93 に、CS-DMA ベースのスキャンモードのフローを示します。

1. Write DMA チャンネル

書き込み DMA は、センサーのスキャン構成を MSC ブロックに転送するように設定されています。対応するセンサーのスキャン設定への送信元アドレスは、Chain Write DMA チャンネルから受信されます。

2. Chain Write DMA チャンネル

MSC ブロックが現在のセンサーのスキャンを完了すると、DMA がトリガーされ、次のセンサーまたは最初のセンサー (新しいスキャンの場合) のスキャン設定の送信元アドレスが Write DMA チャンネルに転送されます。

3. Read DMA チャンネル

Read DMA チャンネルは、スキャン結果 (rawcount) に対応するセンサーの宛先の場所に転送します。

4. Chain Read DMA チャンネル

現在のセンサースキャンが MSC ブロックによって完了すると、Chain Read DMA がトリガーされ、現在のセンサースキャン結果 (rawcount) の宛先の場所 (address) が Read DMA チャンネルに転送されます。

CAPSENSE™性能のチューニング

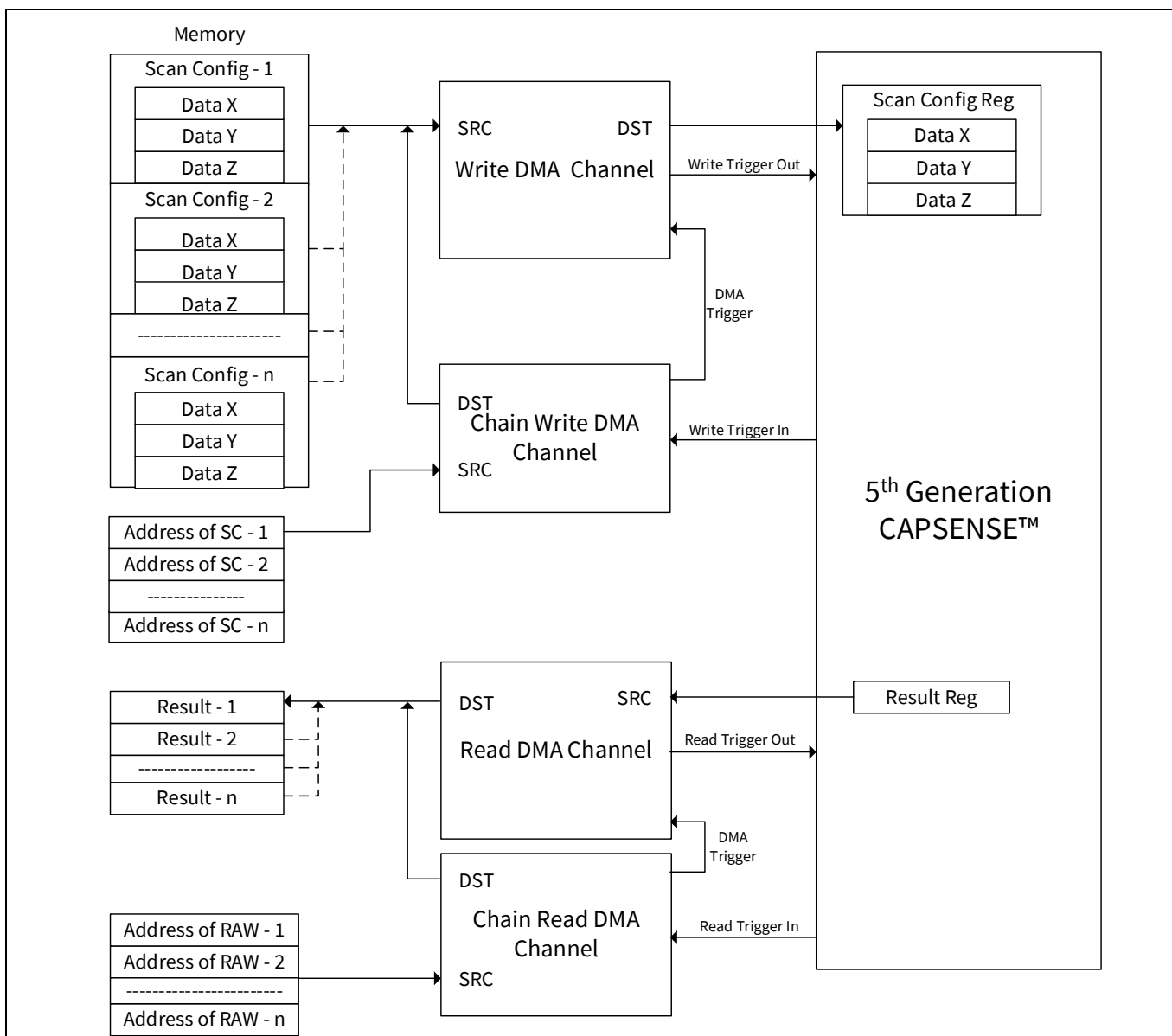


Figure 93 CS-DMA スキャンニングフロー

5.3.4.5 マルチチャネルスキャンニング

マルチチャネル設計では、CAPSENSE™ MSC0 と MSC1 の両方のインスタンスを使用するため、同時操作とスキャン時間の短縮が実現します。マルチチャネルスキャンはロックステップにあるため、クロスチャネルノイズ結合が回避されます。スキャンをロックステップにするには、スキャン同期が必要です。第5世代 CAPSENSE™テクノロジーにはマルチチャネル同期機能が組み込まれており、これには CPU は必要ありません。

マルチチャネル操作は、インターフェースに多くのセンサーピンを必要とする大型タッチパッドなどのアプリケーションをサポートするための追加の利点です。例えば、6x8 タッチパッドは Figure 94 に示すように構成できます。この図では、青色で表示されたセンサーはチャンネル 0 (MSC0) でスキャンされ、緑色はチャンネル 1 (MSC1) でスキャンされます。

CAPSENSE™性能のチューニング

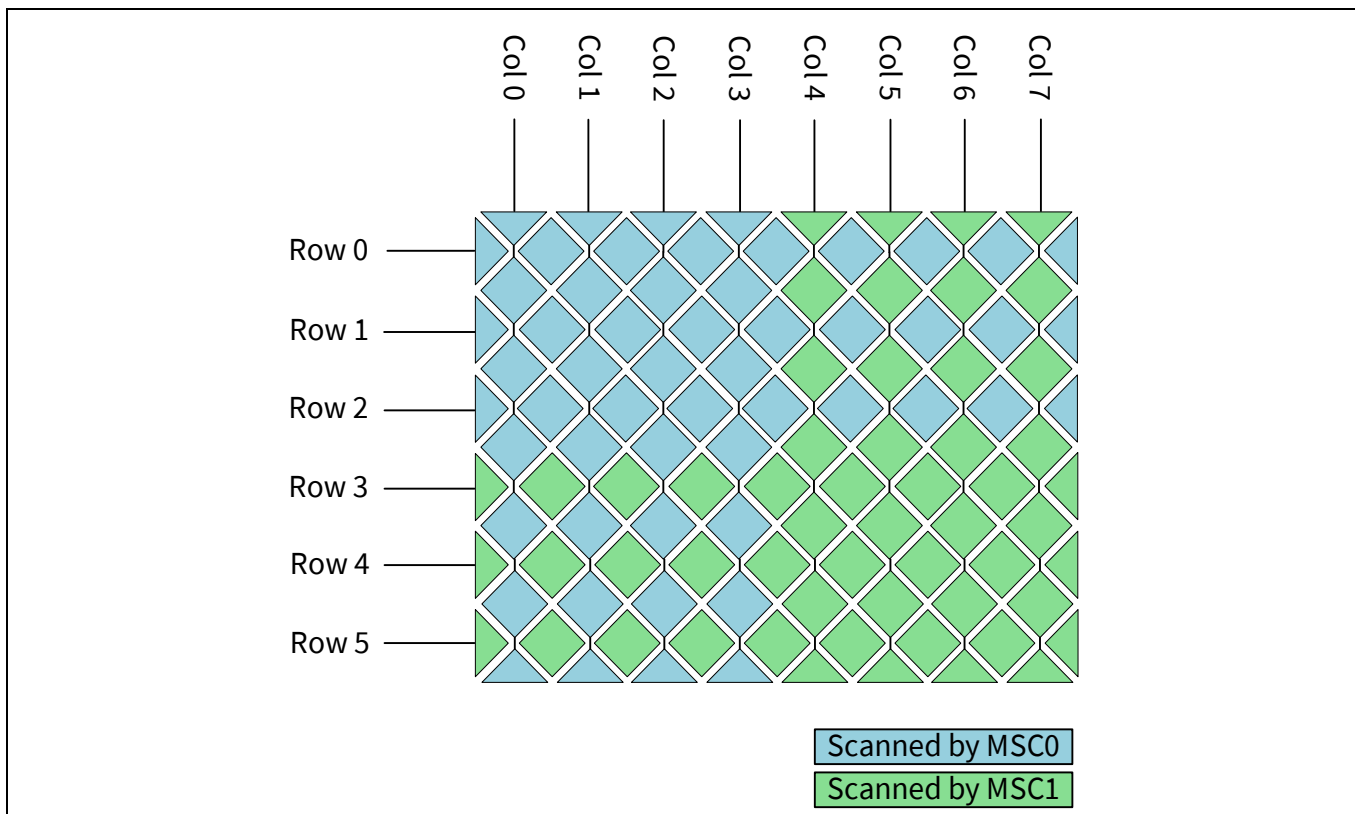


Figure 94 マルチチャンネルを使用した 6x8CSD タッチパッドのスキャン

この場合、チャンネル0とチャンネル1は、そのセンサーの1つを同時にスキャンできます。クロストークノイズを回避するには、2つのセンサー間の物理的距離が可能な限り最大になるように一緒にスキャンするセンサーを選択し、行センサーと列センサーを組み合わせないようにする必要があります。

上記の例では、推奨されるスキャン構成は **Table 17** のとおりです。同じスロットに属するすべてのセンサーが一緒にスキャンされます。

Table 17 チャネルスキャン設定

スロット#	チャンネル0センサー	チャンネル1センサー
Slot 0	Col 0	Col 4s
Slot 1	Col 1	Col 5
Slot 2	Col 2	Col 6
Slot 3	Col 3	Col 7
Slot 4	Row 0	Row 3
Slot 5	Row 1	Row 4
Slot 6	Row 2	Row 5

5.3.4.6 ボタンウィジェット調整

ボタンウィジェット調整セクションでは、CSD ボタンを調整するための高レベルの手順を提供します。**CE231078 PSoc™ 4: MSC CAPSENSE™ CSD Button Tuning** では、ModusToolbox™用の Eclipse IDE での自己静電容量ベースのボタンウィジェットのチューニングについて説明します。コンポーネントおよび

CAPSENSE™性能のチューニング

関連するすべてのパラメーターの詳細については、[コンポーネントのデータシート](#)を参照してください。

5.3.4.7 スライダーウィジェット調整

スライダーウィジェット調整セクションでは、CSD スライダーを調整するための高レベルの手順を提供します。[CE232776 PSoC™ 4: MSC CAPSENSE™ CSD slider tuning](#) では、ModusToolbox™用の Eclipse IDE での自己容量ベースのスライダーウィジェットの調整について説明します。コンポーネントおよび関連するすべてのパラメーターの詳細については、[コンポーネントのデータシート](#)を参照してください。

5.3.4.8 タッチパッドウィジェット調整

タッチパッドウィジェット調整セクションでは、CSD-RM タッチパッドを調整するための高レベルの手順を提供します。[CE232273 PSoC™ 4: MSC Self-capacitance touchpad tuning](#) では、ModusToolbox™用の Eclipse IDE での自己容量ベースのタッチパッドウィジェットの調整について説明します。コンポーネントおよび関連するすべてのパラメーターの詳細については、[コンポーネントのデータシート](#)を参照してください。

マルチチャンネルで CSD-RM タッチパッドウィジェットを使用するための Scan Order タブの基本的なルールは次のとおりです。

1. 第 5 世代 CAPSENSE™でのスキャンは、スロット番号を使用して注文されます。単一のスロット番号をすべてのチャンネルの 1 つのセンサーに割り当て、その特定のスロットをスキャンして、そのスロットのすべてのセンサーを同期してスキャンできます。
2. CSD-RM タッチパッドの場合、同じスロットを行または列にのみ割り当てる必要があります。したがって、クロストークの原因となる行要素と列要素を一緒にスキャンすることは避けてください。
3. スロット番号は、同じスロット番号を持つセンサー間に最大距離が存在するように割り当てる必要があります。
4. 1 つのスロットに CSD センサーと CSX センサーを混在させないでください。
5. タッチパッドセンサーは、スキャン時間を最適化するためにチャンネル間で均等に分割する必要があります。
6. 「コンセンサス」方式が機能するには、すべてのチャンネルに同数のセンサー (スキャン) が必要です。各チャンネルのセンサーの数が等しくない場合、「空のスロット」がそれぞれのチャンネルに追加されます。
7. スロット内では、すべてのセンサーが同じセンスクロックと同じ数のサブ変換を備えている必要があります。

Figure 95 に、8x6 CSD-RM タッチパッドのスロット設定の例を示します。

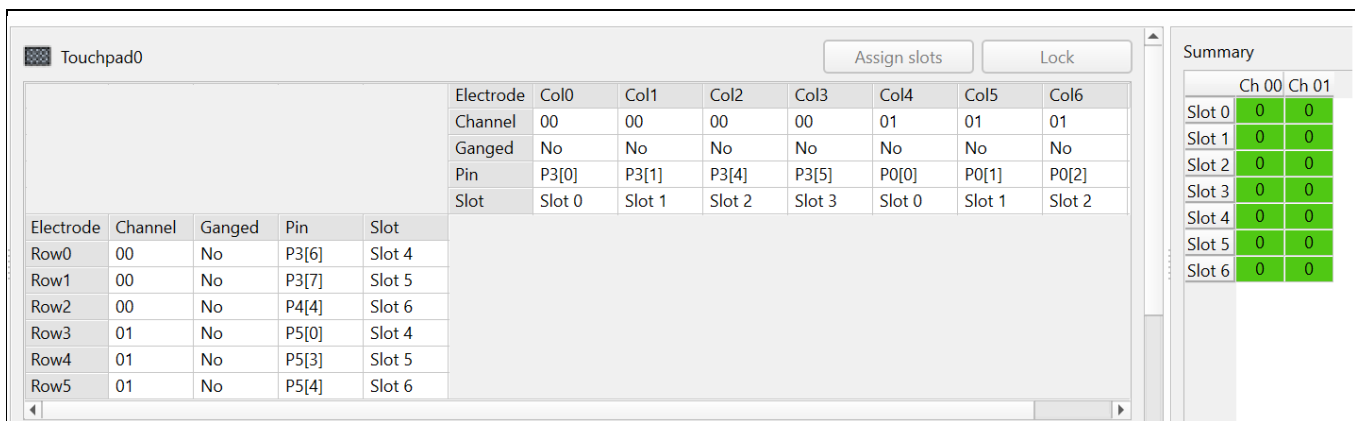


Figure 95 8x6 CSD-RM タッチパッドのスロット設定

CAPSENSE™性能のチューニング

5.3.4.9 近接ウィジェットの例

近接センサーの調整については、[AN92239 - Proximity sensing with CAPSENSE™](#)を参照してください。

5.3.5 CSX-RM センシング方式 (第 5 世代)

ここでは、第 5 世代デバイスの CSX-RM センシング方式を使用した手動チューニングの基本について説明します。また、手動チューニング手順に影響を与えるハードウェアパラメーターについても説明します。

5.3.5.1 基本

5.3.5.1.1 変換利得と CAPSENSE™信号

変換利得は、センサーに指で触れたときにシステムが監視する信号数に影響します。より多くの利得がある場合、信号はより高くなり、より高い信号はより高い達成可能な[信号対ノイズ比 \(SNR\)](#)を意味します。利得を上げると、信号とノイズの両方が増える可能性があることに注意してください。ただし、必要に応じて、ファームウェアフィルターを使用してノイズを減らせます。使用可能なファームウェアフィルターの詳細については、[Table 7](#)を参照してください。

シングル CDAC での変換利得

相互容量検知システムでは、[式 62](#)に示すように、rawcount カウンターは Tx 電極と Rx 電極間の相互容量に正比例します。

式 62. センサー容量に対する raw カウントの関係

$$\text{Rawcount}_{\text{Counter}} = G_{\text{CSX}} C_M$$

ここで、

G_{CSX} = CAPSENSE™ CSX の容量からデジタルへの変換利得

C_M = Tx 電極と Rx 電極間の相互容量

[Figure 97](#)に、CSX センサーの raw カウントと相互容量の関係を示します。[式 63](#)の変換利得の調整可能なパラメーターは、 C_{ref} 、 $\text{TxClk}_{\text{Div}}$ 、および N_{Sub} です。

この変換利得の概算値は次のとおりです。

式 63. 静電容量からデジタルコンバーターの利得

$$G_{\text{CSX}} = \text{MaxCount} \cdot \frac{2}{C_{\text{ref}} \text{TxClk}_{\text{Div}}}$$

ここで、 $\text{MaxCount} = N_{\text{Sub}} * \text{TxClk}_{\text{Div}}$

[式 62](#) および [式 63](#) による、単一 CDAC モードでの raw カウントの式を[式 64](#)に示します。

式 64. シングル CDAC モード raw カウント

$$\text{Rawcount}_{\text{Counter}} = N_{\text{Sub}} \frac{2 * C_M}{C_{\text{ref}}}$$

ここで、

N_{Sub} = サブ変換数

CAPSENSE™性能のチューニング

$TxClk_{Div}$ = Tx クロック分周器

C_M = センサーの相互容量

$C_{ref} = RefCDACCode * C_{lsb}$

RefCDACCode = 基準 CDAC 値

$C_{lsb} = 8.86 \text{ fF}$

変換利得の調整可能なパラメーターは、 C_{ref} , $TxClk_{Div}$, および N_{Sub} です。

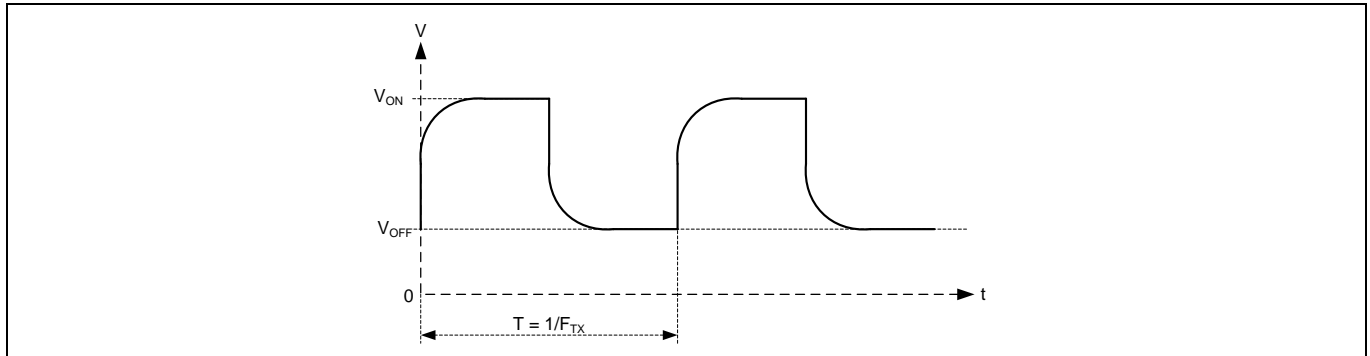


Figure 96 CSX センサーの Tx ノードの電圧

コンポーネントから観測された raw カウントは式 65 で与えられることに注意してください。Rawcount_{component} の詳細については、CAPSENSE™ CSX-RM センシング方式 (第 5 世代) を参照してください。

式 65. Rawcount_{component}

$$Rawcount_{Component} = MaxCount - Rawcount_{Counter}$$

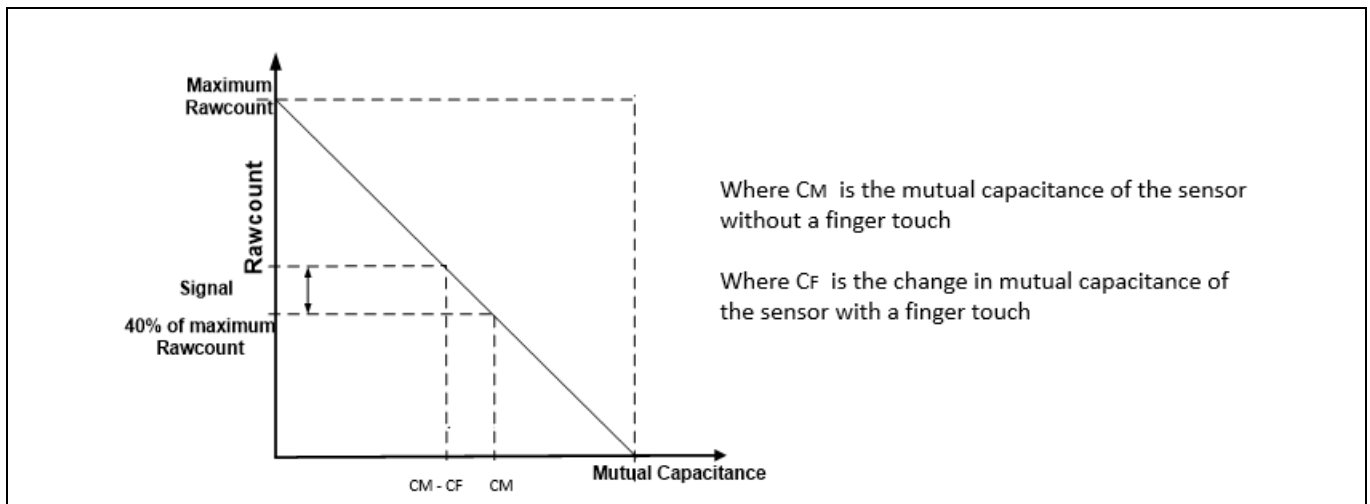


Figure 97 raw カウントとセンサーの相互容量

デュアル CDAC モードでの変換利得

式 23 および式 62 による、デュアル CDAC モードでの raw カウントの式を式 66 に示します。

CAPSENSE™性能のチューニング

式 66. デュアル CDAC モード raw カウント

$$\text{raw カウント} = G_{\text{CSX}} C_M - \text{Maxcount} \frac{2 * C_{\text{comp}}}{C_{\text{ref}} \text{CompCLK}_{\text{div}}}$$

ここで、

$$\text{Maxcount} = N_{\text{Sub}} * \text{SnsClk}_{\text{Div}}$$

$\text{SnsClk}_{\text{Div}}$ = センスクロック分周器

N_{Sub} = サブ変換数

$$C_{\text{ref}} = \text{基準静電容量} = \text{RefCDACCode} * C_{\text{lsb}}$$

$$C_{\text{comp}} = \text{補償静電容量} = \text{CompCDACCode} * C_{\text{lsb}}$$

$\text{CompCLK}_{\text{Div}}$ = CDAC 補償分周器

C_M = センサーの相互容量

RefCDACCode = 基準 CDAC 値

CompCDACCode = 補償 CDAC 値

$$C_{\text{lsb}} = 8.86\text{fF}$$

G_{CSX} は式 63 で与えられます。

5.3.5.2 CAPSENSE™ハードウェアパラメーターの選択

CAPSENSE™ハードウェアパラメーターは、変換利得と CAPSENSE™信号を制御します。Table 18 に、第 5 世代デバイスの CSX-RM センシング方式に適用される CAPSENSE™ハードウェアパラメーターを示します。

Table 18 CAPSENSE™コンポーネントのハードウェアパラメーター

No	ModusToolBox™での CAPSENSE™パラメーター
1	Tx クロック分周器
2	Tx クロックソース
3	変調器クロック分周器
4	基準 CDAC 値
5	CDAC 補償分周器
6	補償 CDAC 値
7	サブ変換数
8	Enable CDAC dither

5.3.5.2.1 スキャンモード

スキャンモードは、CS-DMA または割込み駆動モードとして設定できます。自律スキャンの場合は DMA モードを選択し、レガシー割込みベースのスキャンの場合は割込み駆動モードを選択します。

CAPSENSE™性能のチューニング

5.3.5.2.2 センサー接続方法

自動スキャンは CTRLMUX 方式でのみ使用できますが、この方式ではサポートされるピンの数が制限されます (サポートされるピンについては [デバイスデータシート](#) を参照してください)。さらに、オンチップ IO ノイズに対する耐性が向上します。割込み駆動モードでより多くのピンをサポートするには、AMUXBUS 方式を選択してください。

CSX センサーの CTRLMUX 接続方法で、VDDA/2 として非アクティブセンサー接続を選択し、Rx ラインの電圧を VDDA/2 に初期化するための最初のセンサースキャンの前に空のスキャンスロットを追加してください。空のスロットを作成する詳細な手順については、[タッチパッドウィジェットのチューニング](#) のサンプルコードを参照してください。

5.3.5.2.3 変調器クロック周波数

式 62 および **式 63** が示すように、変調器のクロック周波数が高いほど、特定の C_M からデジタルカウントへの変換の感度/信号が高くなり、精度が高くなり、ノイズが低くなるため、特定のデバイスに許容される最高のクロック周波数を選択するのが最適です。また、 F_{MOD}/F_{TX} の値を高くすると、 C_M から raw カウントへの変換で [フラットスポット](#) の幅が狭くなります。

5.3.5.2.4 初期化サブ変換

初期化の一環として、CMOD は必要な電圧 (VDDA/2) で充電する必要があります。初期化には、 C_{MOD} 初期化、 C_{MOD} ショート、および初期化サブ変換の 3 つのフェーズがあります。 C_{MOD} の初期化フェーズでは、 C_{MOD1} が GND にプルされ、 C_{MOD2} が VDDA にプルされます。 C_{MOD} の短絡フェーズでは、両方のコンデンサが相互に接続されるため、電荷が共有されて、両方で VDDA/2 に近い電圧が生成されます。2 つのフェーズの後、スキャンが開始されますが、初期化サブ変換数のために rawcount は破棄されます。

初期化サブ変換数は、**式 67** に基づいて選択する必要があります。

式 67. 初期化サブ変換数

$$\text{初期化サブ変換数} = \text{ceiling} \left(\frac{C_{MOD} * V_{OS}}{2 * VDDA * C_M * (1 - \text{Base}\%) * \left(\frac{1}{\text{Bal}\%} - 1\right)} \right) + 1$$

または

$$\text{初期化サブ変換数} = \text{ceiling} \left(\frac{C_{MOD} * V_{OS}}{VDDA * \text{TxClkDiv} * C_{ref} * (1 - \text{Bal}\%)} \right) + 1$$

ここで、

C_{MOD} = 変調器コンデンサ

V_{OS} = コンパレータオフセット電圧 (PSoC™ 4100S Max デバイスの 3mV)

C_M = センサーの相互容量

Base% = ベースライン補償率

Bal% = Rawcount キャリブレーション率

TxClkDiv = Tx クロック分周器

C_{ref} = 基準静電容量 = $\text{RefCDACCode} * C_{lsb}$

CAPSENSE™性能のチューニング

RefCDACCode = 基準 CDAC 値

$C_{lsb} = 8.86fF$

5.3.5.2.5 Tx クロックパラメーター

Tx クロックに関連する 2 つのパラメーターがあります。センスクロックソースとセンスクロック周波数です。

Tx クロックソース

コンポーネントのクロックソースとして「自動」を選択すると、ウィジェットごとにダイレクトスペクトルクロック (SSCx) とスペクトラム拡散クロック (SSCx) の間で最適な Tx クロックソースが自動的に選択されます。「自動」オプションが選択されていない場合は、以下に基づいてクロックソースを選択します。

- 直接-固定クロック周波数のクロック信号。ほとんどの場合、このオプションを使用してください。
- スペクトラム拡散クロック (SSCx) - このオプションを選択すると、Tx クロック信号の周波数が所定の範囲に動的に拡散されます。このオプションを使用して、EMI 干渉を減らし、フラットスポットを回避します。
ただし、SSCx クロックを選択するときは、SSCx クロックソース選択の [コンポーネントデータシート / ModusToolbox™ CAPSENSE™ configurator guide](#) に記載されている条件が満たされるように、Tx クロック周波数、変調器クロック周波数、およびサブ変換の数を選択してください。
- 疑似ランダムシーケンス (PRStx) - PRStx (疑似ランダムシーケンス) モードを使用して、フラットスポットを削除し、EMI / EMC 放射と感受性を改善します。第 5 世代 CAPSENSE™ では、PRStx クロックはより高い rawcount キャリブレーションパーセントで信号/感度損失をもたらすため、PRStx クロックを使用する場合は 65% rawcount キャリブレーションが推奨されます。

Tx クロック周波数

Tx クロック周波数は、[CAPSENSE™ CSX-RM センシング方式 \(第 5 世代\)](#) のセクションで説明されているように、各サブ変換の期間を決定します。Tx クロック信号は、センサーの寄生容量を完全に充電および放電する必要があります。オシロスコープで信号をチェックするか、[式 62](#) を使用して設定できます。さらに、自動校正された CDAC コードが選択した F_{TX} のミッドレンジ (例えば、6-200) にあることを確認します。自動キャリブレーションされた CDAC コードが推奨範囲外にある場合は、推奨範囲内に収まり、[式 68](#) を満たすように F_{TX} を調整します。

式 68. Tx クロック周波数を選択するための条件

$$F_{TX} < \frac{1}{2 * 5 * R_{SeriesTx} C_{PTx}}$$

ここで、

C_{PTx} = Tx 電極の寄生容量

$R_{SeriesTx}$ = 内部スイッチの $R_{internal}$ 抵抗を含む総直列抵抗。2kΩ の外部直列抵抗 (センサーパッドをデバイスピンに接続する PCB 配線に接続) を推奨。そして高抵抗材料 (例: ITO または導電性インク) を使用する場合は配線抵抗。

$R_{internal}$ = 内部抵抗。これはスキャンモードによって異なります。[Table 19](#) を参照してください。

CAPSENSE™性能のチューニング

Table 19 センサーの内部抵抗

スキャンモード	R _{internal}
CTRLMUX	950 Ω
AMUXBUS	500 Ω

C_pの値は、CSD ビルトインセルフテスト API を使用して見積もることができます。詳細については、[コンポーネントデータシート/ミドルウェア文書](#)を参照してください。

スキャン時間を最小限に抑えるために、[式 69](#) に示すように、[式 68](#) を満たすコンポーネントドロップダウンリストで使用可能な最大 Tx クロック周波数を使用することを推奨します。

式 69. CSX センサーのスキャン時間

$$T_{CSX} = \frac{N_{Sub}}{F_{TX}}$$

ここで、

N_{Sub} = [サブ変換数](#)

さらに、SSCx クロックソースを使用している場合は、これらの条件に加えて、[コンポーネントデータシート/ミドルウェア文書/ ModusToolbox™ CAPSENSE™ configurator guide](#) に記載される条件を満たす Tx クロック周波数を選択していることを確認してください。

5.3.5.2.6 サブ変換数

サブ変換数によって、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。固定変調器クロックと Tx クロックの[式 23](#) から、サブ変換 (N_{Sub}) の数を増やすと、信号と SNR が増加します。ただし、サブ変換数を増やすと、[式 69](#) に従ってセンサーのスキャン時間も長くなります。

最初に、値を低い数値 (例えば、20) に設定し、チューナー GUI を使用してセンサーの SNR を見つけます。選択した N_{Sub} で SNR が 5:1 を超えていない場合は、SNR 要件が満たされるように段階的に N_{Sub} を増やしてください。

5.3.5.2.7 容量性 DAC

第 5 世代の CSX-RM は、[Figure 49](#) に示すように、C_{MOD} のバランスをとる基準 CDAC (C_{ref}) と補償 CDAC (C_{comp}) の 2 つの CDAC に対応します。これらは、容量からデジタルへの変換のための[変換利得と CAPSENSE™信号](#)を制御します。CAPSENSE™コンポーネントを使用すると、CDAC の次の設定が可能になります。

- 補償 CDAC の有効化または無効化
- CDAC の自動キャリブレーションの有効化または無効化
- 補償 CDAC 分周器、自動キャリブレーションが無効になっている場合の基準および補償 CDAC の DAC コード選択

基準 CDAC (C_{ref})

基準 CDAC は、C_{MOD} からセンサー相互容量 (C_M) によって転送された電荷を補償するために使用されます。切り替え回数は、センサーの相互容量によって異なります。

C_{ref} は以下の基準を満たす必要があります。

CAPSENSE™性能のチューニング

- 補償が無効の場合:
RefCDACCode \geq 6
- 補償が有効な場合:

$$\text{RefCDACCode} \geq \frac{10}{\text{CDAC Compensation Divider}}$$

ここで、

$$C_{\text{ref}} = \text{基準静電容量} = \text{RefCDACCode} * C_{\text{lsb}}$$

$$\text{RefCDACCode} = \text{基準 CDAC 値}$$

$$C_{\text{lsb}} = 8.86\text{fF}$$

補償 CDAC (C_{comp})

補償 CDAC を有効にすることは「デュアル CDAC」モードと呼ばれ、[変換利得と CAPSENSE™信号](#)で説明されているように信号が増加します。ほとんどの場合、補償 CDAC を有効にします。

補償コンデンサは、センサーからの過剰な相互容量を補償して感度を上げるために使用されます。切り替えられる回数は、ユーザーアプリケーションがセンサーの相互容量から補償しようとしている電荷の量によって異なります。

C_{comp} は、以下の基準を満たす必要があります。

- RefCDACCode = 1の場合、CompCDACCode \geq 98

ここで、

$$C_{\text{comp}} = \text{補償静電容量} = \text{CompCDACCode} * C_{\text{lsb}}$$

$$\text{CompCDACCode} = \text{補償 CDAC 値}$$

$$C_{\text{lsb}} = 8.86\text{fF}$$

5.3.5.2.8 補償 CDAC 分周器

シングルセンスクロックで補償コンデンサが切り替わる回数を K_{comp} で表します。以下の基準が満たされるように、以下の[式 70](#)に基づいて CDAC 補償分周器を選択します。

- CDAC 補償分周器 \geq 4
- K_{comp} は整数である必要があります。

式 70. CDAC 補償分周器

$$\text{CDAC compensation divider} = \frac{\text{Tx clock divider}}{K_{\text{comp}}}$$

5.3.5.2.9 自動キャリブレーション

この機能により、ファームウェアは CDAC を自動的にキャリブレーションして、必要なキャリブレーション目標である 40%を達成できます。ほとんどの場合、自動キャリブレーションを有効にすることを推奨します。この機能を有効にすると、次のようになります。

- パーツ間の C_M の変動があっても、raw カウントのキャリブレーションを最大 raw カウントの 40%に修正
- [フラットスポット](#)の影響を減らす

CAPSENSE™性能のチューニング

- 最適な利得を自動的に選択

さまざまな環境条件下で CAPSENSE™を適切に機能させるために、非常に低いまたは高い CDAC コードを回避することを推奨します。CAPSENSE™チューナーを使用して、自動キャリブレーションされた CDAC 値がこの推奨範囲内にあることを確認できます。CDAC 値が [式 62](#), [式 63](#), および [式 65](#) に基づく推奨範囲外の場合は、キャリブレーションレベルまたは F_{mod} または F_{sw} を変更して、CDAC コードを適切な範囲に収めることができます。

5.3.5.2.10 CDAC コードの選択

これは推奨されるアプローチではありません。ただし、これは、何らかの理由で自動キャリブレーションを無効にする場合にのみ使用できます。CDAC コードを取得するには、最初に自動キャリブレーションを有効にして CAPSENSE™コンポーネントを設定し、他のすべてのハードウェアパラメーターを最終調整に必要なものと同じにして、[チューナーGUI](#) を使用してキャリブレーションされた CDAC 値を読み戻します。次に、CAPSENSE™コンポーネントを再設定して自動キャリブレーションを無効にし、取得した CDAC コードを [チューナーGUI](#) から読み戻される固定 DAC コードとして使用します。

5.3.5.2.11 CDAC ディザ

入力容量がスイープされると、raw カウントは容量とともに直線的に増加するはずですが、raw カウントが入力容量によって線形に変化しない領域があります。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、フラット領域の周りで変換ポイントを移動するホワイトノイズを追加します。

5.3.5.3 CAPSENSE™ソフトウェアパラメーターの選択

第 5 世代の CAPSENSE™ソフトウェアパラメーターは、第 4 世代のパラメーターと同じです。したがって、これらのパラメーターは、[CAPSENSE™ソフトウェアパラメーターの選択](#)セクションで説明したように選択できます。

5.3.5.4 自律スキンの構成

CSX-RM センシングでの自律スキンの構成は、CSD-RM センシングの構成と同じです。したがって、[自律スキンの設定](#)セクションで説明されているように、自律スキンを構成します。

5.3.5.5 マルチチャネルスキン

CSX-RM センシングのマルチチャネルスキンは、CSD-RM センシングのマルチチャネルスキンと同じです。したがって、詳細については、[マルチチャネルスキニング](#)セクション [5.3.4.5](#) を参照してください。

5.3.5.6 ボタンウィジェットの調整

[ボタンウィジェット調整](#)セクションでは、CSX ボタンを調整するための高レベルの手順を提供します。[CE231079 PSoC™ 4: MSC CAPSENSE™ CSX button tuning](#) では、ModusToolbox™用の Eclipse IDE での相互容量ベースのボタンウィジェットの調整について説明します。コンポーネントおよび関連するすべてのパラメーターの詳細については、[コンポーネントのデータシート](#)を参照してください。

CAPSENSE™性能のチューニング

5.3.5.7 タッチパッドウィジェットのチューニング

タッチパッドウィジェット調整セクションでは、CSX タッチパッドを調整するための高レベルの手順を提供します。CE232275 PSoC™ 4: MSC multi-touch mutual-CAPSENSE™ touchpad tuning は、ModusToolbox™用の Eclipse IDE での相互容量ベースのボタンウィジェットのチューニングについて説明しています。コンポーネントおよび関連するすべてのパラメーターの詳細については、コンポーネントのデータシートを参照してください。

マルチチャンネルが有効になっている場合の CSX ウィジェットの Scan Order タブのルール:

1. 第5世代 CAPSENSE™でのスキャンは、スロット番号を使用して注文されます。シングルスロット番号をすべてのチャンネルの1つのセンサーに割り当て、その特定のスロットをスキャンして、そのスロットのすべてのセンサーを同期してスキャンできます。
2. スロット番号は、同じスロット番号を持つ Rx 電極間に最大距離があり、潜在的なクロストークを回避するように割り当てる必要があります。
3. センサーの Tx および Rx 電極は、2つの異なるチャンネルまたは同じチャンネルに割り当てることができます。センサーは、センサー Rx 電極が接続されているチャンネルに属しています。
4. スキャン期間を最適化するために、Rx 電極をチャンネル間で均等に分割する必要があります。
5. どのチャンネルでも、すべてのチャンネルの Tx 信号を生成できます。
6. Tx 電極は、チャンネル間で任意の順序で割り当てることができます。
7. 「コンセンサス」方式が機能するには、すべてのチャンネルに同数のセンサー (スキャン) が必要です。各チャンネルのスキャン数が等しくない場合、「空のスロット」がそれぞれのチャンネルに追加されます。
8. CSD センサーと CSX センサーを1つのスロットに混在させないでください。
9. スロット内では、すべてのセンサーが同じセンスクロックと同じ数のサブ変換を備えている必要があります

Figure 98 に、8x6 CSX-RM タッチパッドのスロット設定の例を示します。

Electrode	Channel	Ganged	Pin	Rx0	Rx1	Rx2	Rx3	Rx4	Rx5	Rx6	Rx7
Tx0	00	No	<none>	00	00	00	00	01	01	01	01
Tx1	00	No	<none>	No	No	No	No	No	No	No	No
Tx2	00	No	<none>	<none>	<none>	<none>	<none>	<none>	<none>	<none>	<none>
Tx3	01	No	<none>	<none>	<none>	<none>	<none>	<none>	<none>	<none>	<none>
Tx4	01	No	<none>	<none>	<none>	<none>	<none>	<none>	<none>	<none>	<none>
Tx5	01	No	<none>	<none>	<none>	<none>	<none>	<none>	<none>	<none>	<none>

Electrode	Channel	Ganged	Pin	Slot 0	Slot 6	Slot 12	Slot 18	Slot 0	Slot 6	Slot 12	Slot 18
Tx0	00	No	<none>	Slot 0	Slot 6	Slot 12	Slot 18	Slot 0	Slot 6	Slot 12	Slot 18
Tx1	00	No	<none>	Slot 1	Slot 7	Slot 13	Slot 19	Slot 1	Slot 7	Slot 13	Slot 19
Tx2	00	No	<none>	Slot 2	Slot 8	Slot 14	Slot 20	Slot 2	Slot 8	Slot 14	Slot 20
Tx3	01	No	<none>	Slot 3	Slot 9	Slot 15	Slot 21	Slot 3	Slot 9	Slot 15	Slot 21
Tx4	01	No	<none>	Slot 4	Slot 10	Slot 16	Slot 22	Slot 4	Slot 10	Slot 16	Slot 22
Tx5	01	No	<none>	Slot 5	Slot 11	Slot 17	Slot 23	Slot 5	Slot 11	Slot 17	Slot 23

Summary		
	Ch 00	Ch 01
Slot 0	0	0
Slot 1	0	0
Slot 2	0	0
Slot 3	0	0
Slot 4	0	0
Slot 5	0	0
Slot 6	0	0
Slot 7	0	0
Slot 8	0	0
Slot 9	0	0
Slot 10	0	0
Slot 11	0	0
Slot 12	0	0
Slot 13	0	0
Slot 14	0	0
Slot 15	0	0
Slot 16	0	0
Slot 17	0	0
Slot 18	0	0

Figure 98 8x6 CSX-RM タッチパッドのスロット設定

CAPSENSE™性能のチューニング

5.3.6 手動チューニングのトレードオフ

設計を手動でチューニングするとき、設定は静電容量センシングシステムの特性に影響を与えることを理解する必要があります。CAPSENSE™設計の3つの主な性能特性は信頼性、消費電力、および応答時間です。

- **信頼性**は CAPSENSE™システムがノイズのある環境または水の存在などの悪条件下でどのように動作するかを定義します。高い信頼性がある設計は誤タッチのトリガーを避け、意図したタッチがこれらの悪条件に登録されることを保証します。
- **消費電力**はデバイスによって消費される平均的な電力として定義されます。これは**低消費電力の設計**に説明するように、スキャン、処理および低消費電力モード遷移を含みます。センサーのより早いスキャンおよび処理は、デバイスが高い電力状態でより短い時間を費やし、低消費電力スリープ状態の時間を最大化することを保証します。
- **応答時間**は指がセンサーにタッチする時点からシステムの応答がある時点までの時間を定義します。最低の応答時間はセンサーのスキャンおよび処理時間で限定されるため、タイミングバジェットを最適に定義し、管理する必要があります。合計応答時間の良い目標は 100ms 以下です。

すべての性能特性は相互に依存します。チューニングプロセスの目的はプロジェクトの特定要件を満たす最適な割合を探すことです。設計を計画するとき、これらの特性は常に逆の関係にあることに注意してください。1つの特性を改善すれば、他の特性が低下します。

例えば、CAPSENSE™をおもちゃに使いたい場合、早い応答時間および低消費電力はより重要になります。オープンの「Start/Stop」ボタンなどの例では、信頼性は最も重要な特性であり、応答時間および消費電力は2番目です。

次に、信頼性、消費電力、および応答時間に影響を与える要因について考えてみましょう。Figure 99 に CAPSENSE™特性の測定可能なパラメーターと実際の CAPSENSE™設定可能なパラメーターとの依存関係を示します。

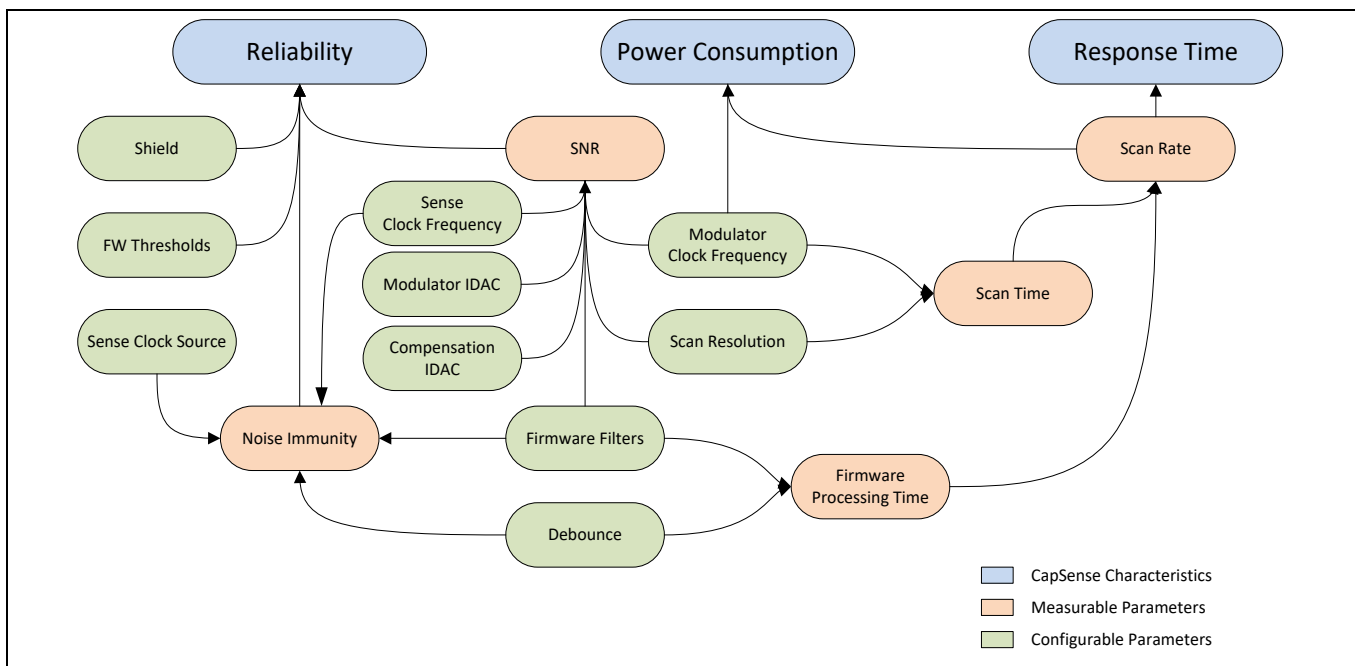


Figure 99 CAPSENSE™パラメーターの関係

CAPSENSE™性能のチューニング

5.3.6.1 信頼性

以下の要因は信頼性に影響をあたえます。

1. 信号対ノイズ比 (SNR)

SNR は有効なタッチ信号の信頼性の尺度を与えます。信頼の CAPSENSE™動作は、SNR が 5 以上であることが必要です。手動チューニングは特定の設計で最適な SNR を確保できます。

2. ノイズ耐性

これは外部または内部のノイズに対向するためのシステム機能です。一般的な外部ノイズの例は、ESD イベントや Bluetooth® LE、リレー スイッチ、電源などの高周波送信です。内部ノイズ源では、例えば、PWM または I2C または SPI 通信による LED 駆動である可能性があります。良好な SNR がある設計でも低いノイズ耐性により性能劣化につながる可能性があります。手動チューニングは、異なるパラメーターを厳密に制御することにより、周波数およびパラメーターを調整し、ノイズ干渉から回避するために役立ちます。

5.3.6.2 消費電力および応答時間

以下の要因は消費電力および応答時間に影響を与えます。

1. スキャン速度

スキャン速度はセンサーのスキャンの周波数として定義されます。スキャン速度は指のタッチからそれを報告されるまでの最小の可能な時間を決めます。最大のスキャン速度は**センサースキャン**によって制限されます。

2. スキャン時間

これは特定センサーをスキャンし、処理する時間です。これは**低消費電力の設計**に示す消費電力および上記に示すスキャン速度に影響を与えます。手動チューニングは最小の SNR を維持しながら、特定のスキャン期間を達成できます。

3. ファームウェア タッチ遅延

これは、使用している CAPSENSE™コンポーネントのバージョンに応じて、**デバウンス**手順または raw データノイズフィルターの使用が原因である可能性があります。これら両方は、センサー処理時間の追加によってスキャン時間に影響を与え、連続で一定のサンプル数がタッチ信号を示すまでタッチ報告を遅延させます。

いずれの場合も、応答速度が低下しますが、信頼性は常に改善されます。

5.3.7 チューニング デバッグ FAQ

ここでは、CAPSENSE™コンポーネント チューニングの一般的なデバッグの問題を提供します。デバッグ問題に応じてさまざまな原因および解決策に関する情報を素早く得るために、あなたが持っている質問にジャンプしてください。

5.3.7.1 チューナーがデバイスと通信しない

原因 1: デバイスはプログラムされていない。

ソリューション 1: チューナーを起動する前に、最新版のプロジェクトを使用してデバイスを**プログラム**してください。

原因 2: チューナーのコンフィギュレーション設定は SCB のコンポーネント設定と一致していない。

CAPSENSE™性能のチューニング

ソリューション 2: EZI2C スレーブ コンポーネント コンフィギュレーション ウィンドウ (「Configure 'SCB_P4」 ダイアログ) を開き、チューナー通信セットアップのダイアログの設定と一致するかを検証してください。チューナーの使用の詳細は [CAPSENSE™コンポーネントのデータシート](#) を参照してください。

原因 3: ユーザーの I2C のピンは正しく設定されていない。

ソリューション 3: ワークスペース エクスプローラで .cydwr ファイルを開き、ピン配置が物理的に基板に接続されているものと一致することを確認してください。

原因 4: CAPSENSE™ TunerStart API やその他の必要とするチューナー コードがない。

ソリューション 4: *main.c* に [CAPSENSE™コンポーネント](#) データシートに記載されたチューナー コードを追加し、デバイスを再プログラムしてください。

5.3.7.2 チューナーを介してデバイスのパラメーターを更新できない

原因 1: デバイスとの通信設定が正しくない。

ソリューション 1: UART/EZI2C configurator ダイアログの設定がチューナー通信セットアップのダイアログの設定と一致することを確認してください。サブアドレスのサイズが同じであることを確認してください。

5.3.7.3 デバイスに接続できますが、raw カウントが表示されない

原因 1: プロジェクトにチューナー コードを追加していない。

ソリューション 1: [チューナー GUI](#) を参照し、*main.c* にチューナー コードを追加してデバイスを再プログラムしてください。

5.3.7.4 差分カウントはセンサーに指を置いた時に、10 から 20 カウントまで少しだけ変化する

原因 1: ユーザーのシステム利得が低すぎる。

ソリューション 1: 本書の [チューナー GUI](#) を参照してください。

原因 2: センサーの寄生容量が非常に高い。

ソリューション 2: この問題を確認するためには、[コンポーネント データシート](#) に記載される組み込みセルフテスト (BIST) の API を使用してください。これらの関数を使用すると、センサーの寄生容量の推定値を読み出せます。また、この読み出しを LCR メーターで独立して確認できます。

ユーザーのハードウェアでは [被駆動シールド信号およびシールド電極](#) を有効にするオプションがあれば、CAPSENSE™コンポーネント構成ウィンドウの詳細設定でこのオプションを使用します。センサー周囲の被駆動シールドは寄生容量を減らすために役立ちます。このオプションを有効にすると、詳細設定で「Inactive Sensorconnection」の設定を「shield」に変更することによって未使用センサーの被駆動シールドを有効にできます。シールドを有効にした後、 C_p が PSoC™デバイスでサポートされている寄生容量の範囲を超えたままの場合は、[PCB レイアウトガイドライン](#) に従ってボードレイアウトを確認して C_p をさらに減らすか、[テクニカル サポート](#) に連絡してレイアウトを確認してください。サポートされている C_p の範囲の詳細については、[コンポーネント データシート/ミドルウェア文書](#) を参照してください。

原因 3: オーバーレイが厚すぎる。

ソリューション 3: ユーザーの [オーバーレイの厚さ](#) に関する [オーバーレイの厚さ](#) を検証してください。

CAPSENSE™性能のチューニング

原因 4: raw カウントが飽和に非常に近い状態であるため、センサーがタッチされると、飽和状態になる。

ソリューション 4: **変調 IDAC および補償 IDAC** に従って IDAC を調整し、raw カウントが特定のセンサーの最大 raw カウントの約 85% に調整してください。

5.3.7.5 システムのチューニングした後、テスト中に大量の放射ノイズがある

原因 1: システム内でセンス クロック周波数が放射ノイズを引き起こしている。

ソリューション 1: **電磁適合性 (EMC) に対する考慮事項** セクションに基づいて、センス クロック周波数を下げるか、またはセンサーの PRS を有効にしてください。既に有効になっている場合は **電磁適合性 (EMC) に対する考慮事項** セクションを参照してください。

原因 2: 大きいシールド電極は大きい放射ノイズに影響を与える可能性がある。

ソリューション 2: **耐液性のレイアウト ガイドライン** に従い、シールド電極のサイズを縮小してください。

5.3.7.6 手動チューニング後、スキャン時間はシステム要件を満たさない

原因: システムのノイズおよび C_p が高いので、信頼できる動作を実現するためにより長いスキャン時間と多くのフィルターが必要です。

ソリューション: C_p を減らす必要があります。まず、CAPSENSE™コンポーネント コンフィギュレーション ウィンドウの詳細設定で **被駆動シールド信号およびシールド電極** を有効にし、**PCB レイアウト ガイドライン** を確認して利得をできるだけ高いレベルに設定することを確保してください。システムはまだ最終的な条件を満たせない場合、**PCB レイアウト ガイドライン** を確認し、さらに C_p を減少させるための基板レイアウトに変更する必要があります。

5.3.7.7 システムは 85% に校正できない

原因 1: センサーはグラウンドに短絡している可能性がある。

ソリューション: 最初にマルチメータを使用してセンサーとグラウンド間の短絡があるかどうかをチェックしてください。短絡があれば、エラーの回路図およびレイアウトを確認してください。

原因 2: センサー C_p は、非常に高い、または非常に低い可能性がある。

ソリューション: ハードウェアに **被駆動シールド信号およびシールド電極** を有効にするオプションがある場合は、CAPSENSE™コンポーネント構成ウィンドウの詳細設定でこのオプションを使用してください。センサーの周りの駆動シールドは、寄生容量を減らすために役立ちます。シールドを使用するハードウェアオプションがない場合、またはシールドを有効にした後も C_p がデバイスでサポートされている C_p よりも大きい場合は、**テクニカルサポート** に連絡してレイアウトを確認するか、アプリケーション固有のガイダンスを確認してください。

デバイスでサポートされている最小寄生容量と比較して容量が低いと思われる場合は、コンデンサのフットプリントをピンに追加してください。最終設計で、 C_p がサポートされている範囲よりも低いと識別された場合は、追加の補償コンデンサを配置して、スキャン中にセンサーに動的に接続することにより、センサー C_p をサポートされている範囲に増やしてください。必要に応じて C_p を上げるために、ピンに接続された外部補償コンデンサにセンサーを接続する方法を理解するためには、**コンポーネント データシート/ミドルウェア文書** を参照してください。

CAPSENSE™性能のチューニング

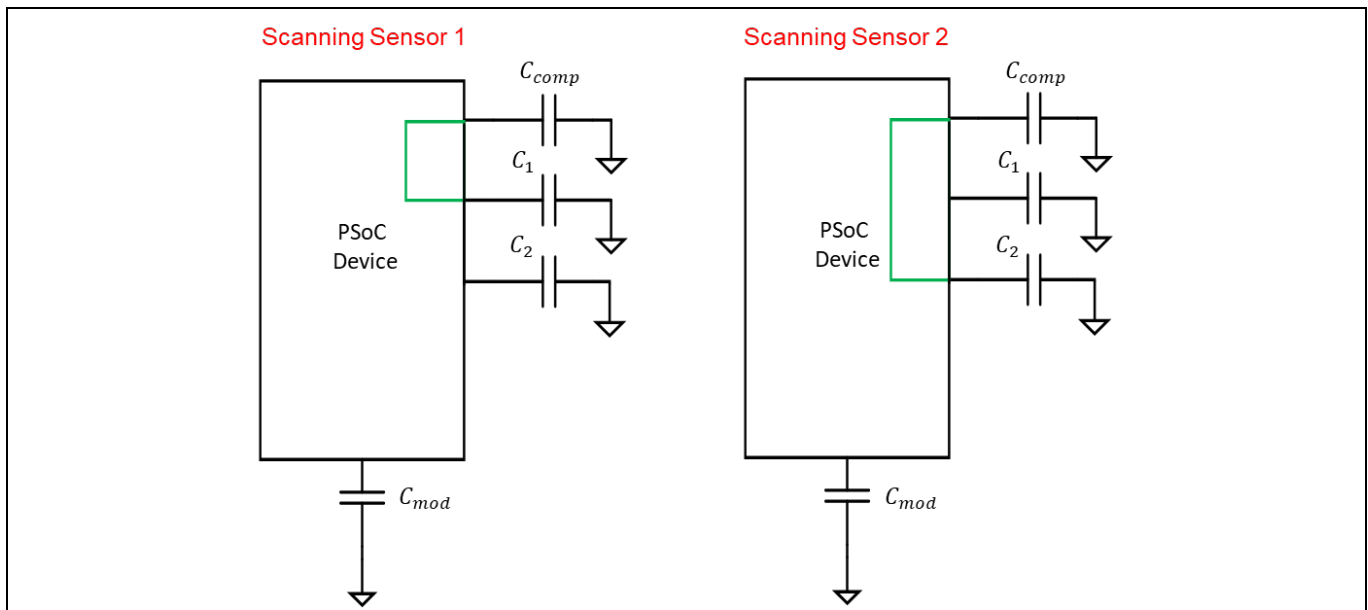


Figure 100 センサーを外部補償コンデンサに接続

5.3.7.8 スライダーの重心応答が非線形である

原因: レイアウトは適切な直線性を確保するために、ハードウェア設計ガイドラインを満たしていない。

ソリューション: CAPSENSE™ コンフィギュレーションウィンドウの General タブで、組み込みセルフテストのオプションを使用してセンサーの C_p をチェックし、[スライダーの設計](#)に従ってレイアウトを更新してください。BIST API の詳細は[コンポーネント データシート/ミドルウェア文書](#)を参照してください。

5.3.7.9 スライダー セグメントは大きな変動のある C_p がある

原因: レイアウトの設計により、センサーに不均衡な C_p がある。

ソリューション: レイアウトを更新する必要があります。[スライダーの設計](#)を参照し、必要に応じてレイアウトを変更してください。すぐに変更できない場合は、同じ応答を受けるために各センサーを再調整する必要があります。これは長い反復プロセスになりますので、できればハードウェアを変更することを推奨します。

5.3.7.10 GPIO を切り替えたとき raw カウントがレベルシフトまたはノイズの増加を示す

原因 1: センサートレースは、PCB 上のトグル GPIO と並列にルーティングされている。

ソリューション: レイアウトを更新する必要があります。[配線の経路](#)を確認し、必要に応じてレイアウトを更新してください。現段階でレイアウトを変更できない場合は、ファームウェアフィルターの使用を評価して、ピークツーピークノイズを低減し、SNR を向上できます。

原因 2: 大量の電流が GPIO を介してシンクされている。

ソリューション: CAPSENSE™センサーがスキャンされているときに GPIO を通過する DC 電流シンクの量を制限してください。[回路図のルールチェックリスト](#)を参照してください。GPIO を介した電流シンクがファームウェア制御されており、電流シンクによって引き起こされる生のカウントレベルシフトがタ

CAPSENSE™性能のチューニング

タッチ信号と比較して大きな違いがある場合、電流シンクが GPIO を介して有効であるときはいつでも、CAPSENSE™ベースラインをリセットまたは再初期化するなどのファームウェア技術を実装できます。以下に示すように、CAPSENSE™センサーのベースラインは CapSense_InitializeWidgetBaseline() API 関数を使用してリセットできます。

```
CapSense_InitializeWidgetBaseline(CapSense_CSD_BUTTON_WDGT_ID);
```

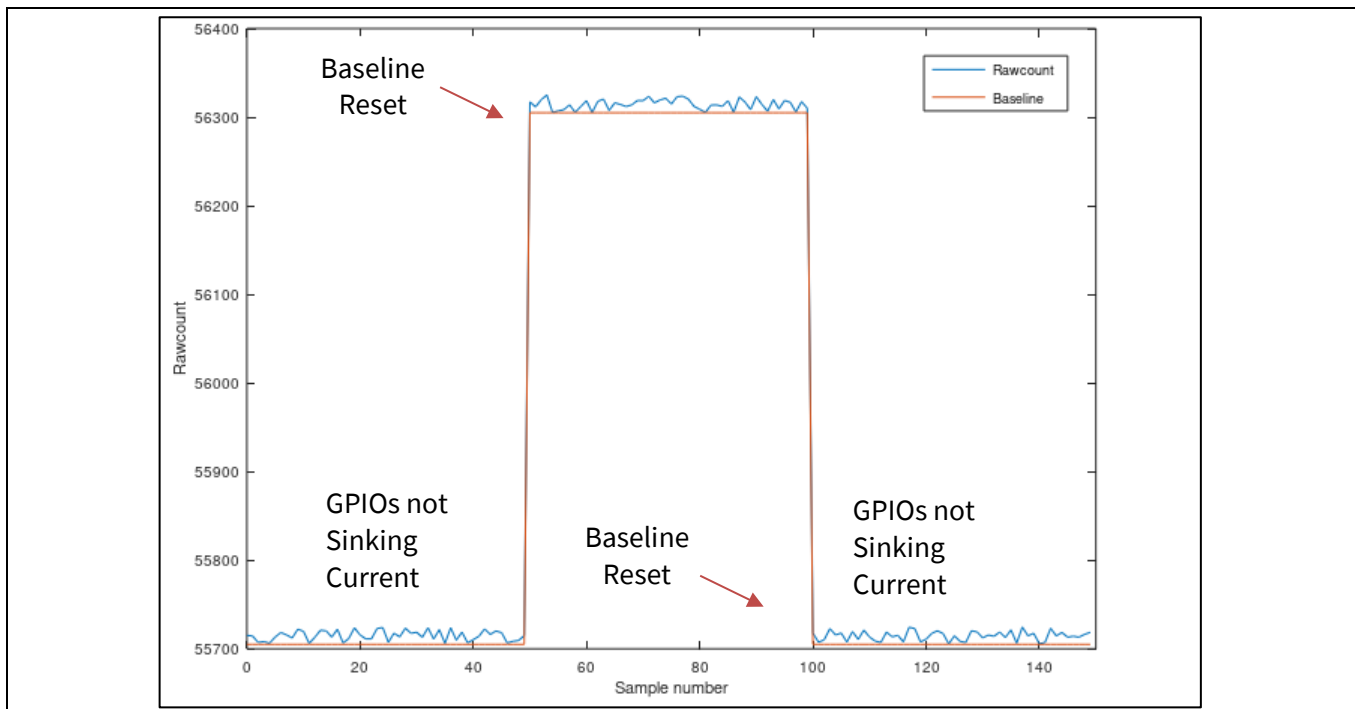


Figure 101 ファームウェア技術を使用したベースラインのリセット

原因 3: センサーピンの選択に記載されているガイドラインに従わなかった。

ソリューション: センサーピンの選択の推奨事項に従ってください。さらに、PSoC™ 6 ファミリのデバイスの場合、ドライブモードの強度、スイッチング周波数、スルーレートの選択などに関する次のガイドラインに従ってください。

- スwitching GPIO のドライブ強度を下げてください。Table 20 に、GPIO で使用可能なドライブ強度オプションを示します。Figure 102 に、ModusToolbox™プロジェクトの Device configurator を使用して GPIO のドライブ強度を選択する方法の例を示します。

Table 20 GPIO の駆動強度

駆動強度	駆動電流 (mA)
Full	8
$\frac{1}{2}$	4
$\frac{1}{4}$	2
$\frac{1}{8}$	1

- トグルする GPIO のスイッチング周波数を下げてください。

CAPSENSE™性能のチューニング

- GPIO スルーレートを SLOW モードとして使用してください (これにより、トグル周波数が 1.5 MHz に制限されることに注意してください)。詳細については、Table 38 を参照してください。
- PRS をセンスクロックソースとして使用してください。
- 可能であれば、VDDA を 2.7V 未満に下げてください。
- GPIO スイッチングを CAPSENSE™ スキャン間の間隔に制限するようにしてください。

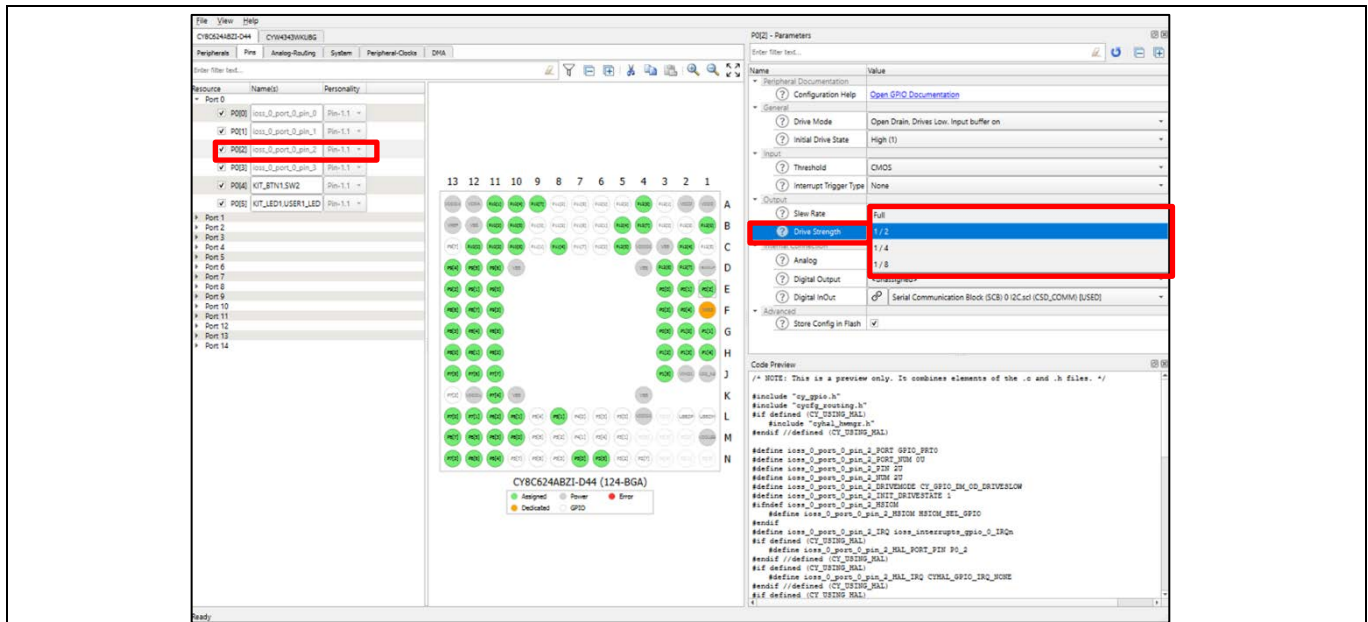


Figure 102 GPIO の駆動強度の選択

5.3.7.11 SNR が低くなっている

原因 1: センサーが適切に調整されていない。

ソリューション: CAPSENSE™性能のチューニングガイドラインに従ってください。

原因 2: CAPSENSE™およびその他の周辺機器が推奨ピンに適切に割り当てられていない。

ソリューション: 詳細については、センサーピンの選択と GPIO を切り替えたとき raw カウントがレベルシフトまたはノイズの増加を示す、を参照してください。

原因 3: HFCLK ソースは、PSoC™ 6 デバイスでより高いノイズを引き起こしている可能性があります。

ソリューション: PSoC™ 6 ファミリー デバイスで CAPSENSE™の最高の性能を得るためには、IMO/ECO+PLL クロックソースから派生した HFCLK を使用してください。このクロックソースは、最高の SNR 性能を提供します。Figure 103 に、ModusToolbox™プロジェクトの Device configurator の System タブを使用してクロック設定を変更する方法を示します。デバイスクロックの変更の詳細については、AN221774 - Getting started with PSoC™ 6 MCU も参照してください。

CAPSENSE™性能のチューニング

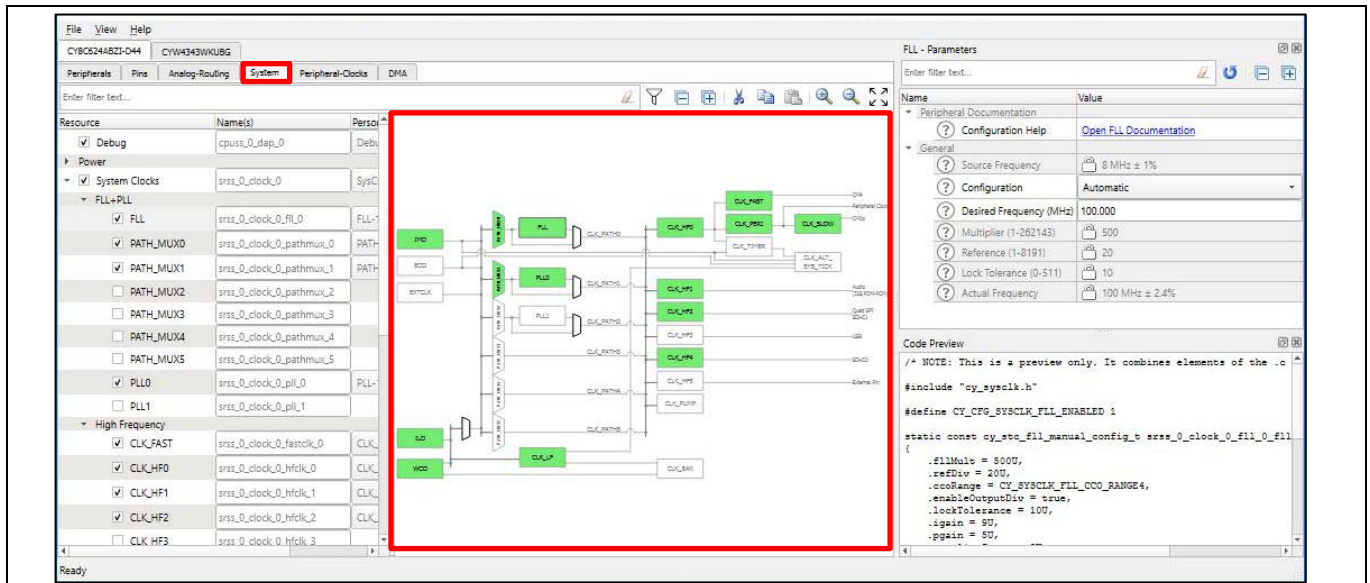


Figure 103 Device configurator でのクロック設定の変更

5.3.7.12 CSX ボタンの C_M が低いことを確認

原因: 適切な IDAC キャリブレーションを行うには、Tx 電極と Rx 電極間の相互容量を約 750fF より大きくする必要があります。

ソリューション: ボタンの C_M が低いことが判明した場合に容量を追加するために、フットプリントのある 2 つの空きピンをデバイスに配置することを推奨します。次に、下の図に示すように、スキャン中に外部コンデンサを CSX センサーに動的に接続することで、センサーの C_M をサポート範囲まで増やせます。ここで、Pin1 はセンサーの Tx ピンに接続され、Pin2 はセンサーの Rx ピンに接続されます。これは、テスト段階で低 C_M リスクが見つかった場合に、それを軽減するのに役立ちます。センサーをギャングする方法を理解するには、[コンポーネントデータシート/ミドルウェア文書](#)を参照してください。

Figure 104 に、CAPSENSE™コンポーネントのボタンウィジェットとして外部コンデンサを追加し、Tx および Rx 電極に専用ピンを割り当てることを示します。**Figure 105** に、選択したピンをセンサーピンと外部コンデンサピンの両方に割り当てることによるセンサーの外部コンデンサへの連動を示します。これは、Rx 電極と Tx 電極の両方に対して行う必要があります。ウィジェットのスキャン中に外部コンデンサをスキャンする必要がないため、CAPSENSE™コンポーネントによって提供される API `CapSense_SetupWidget()` および `CapSense_Scan()` を使用してウィジェットを選択的にスキャンできます。

CAPSENSE™性能のチューニング

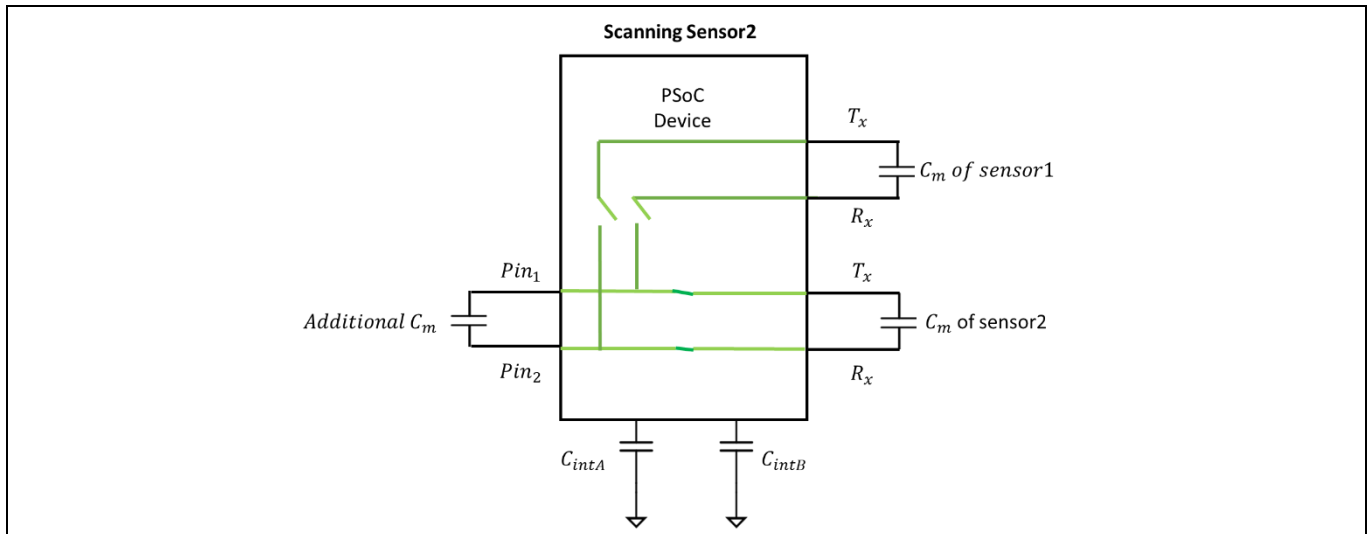


Figure 104 センサーの C_m を増加させるための外部コンデンサのギャング

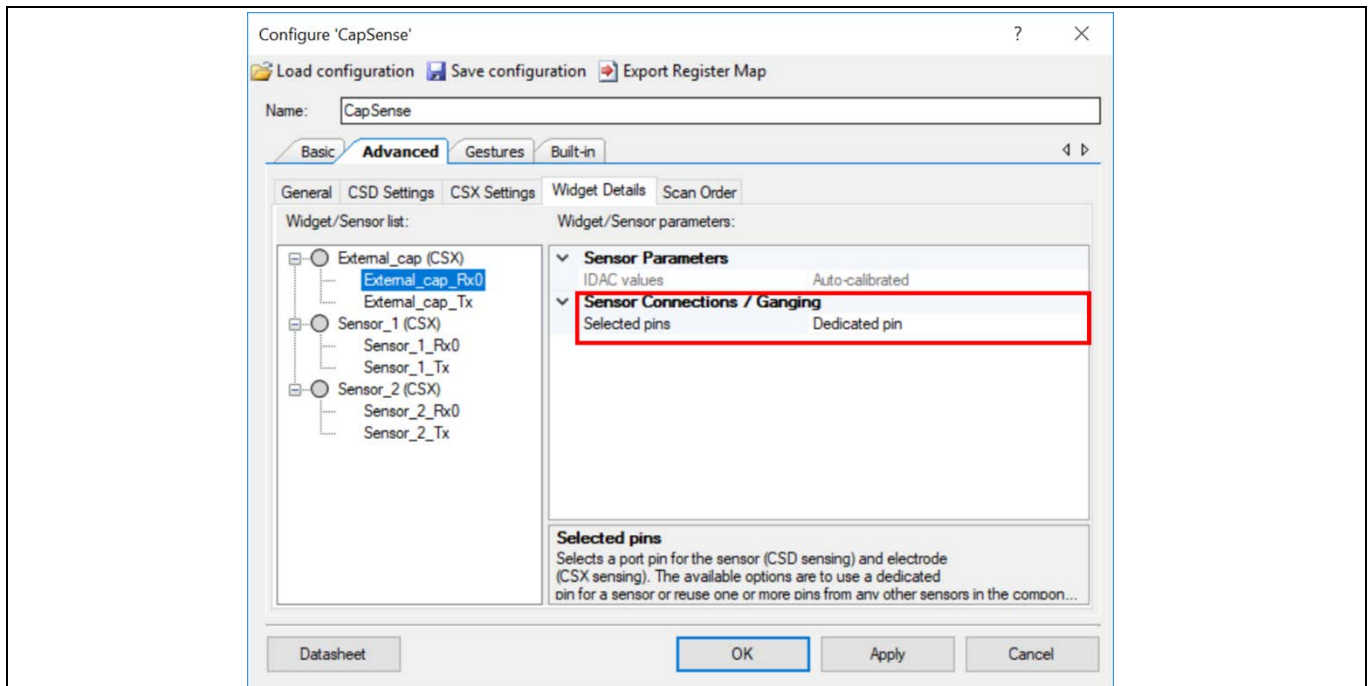


Figure 105 外部コンデンサへの専用ピンの割り当て

CAPSENSE™性能のチューニング

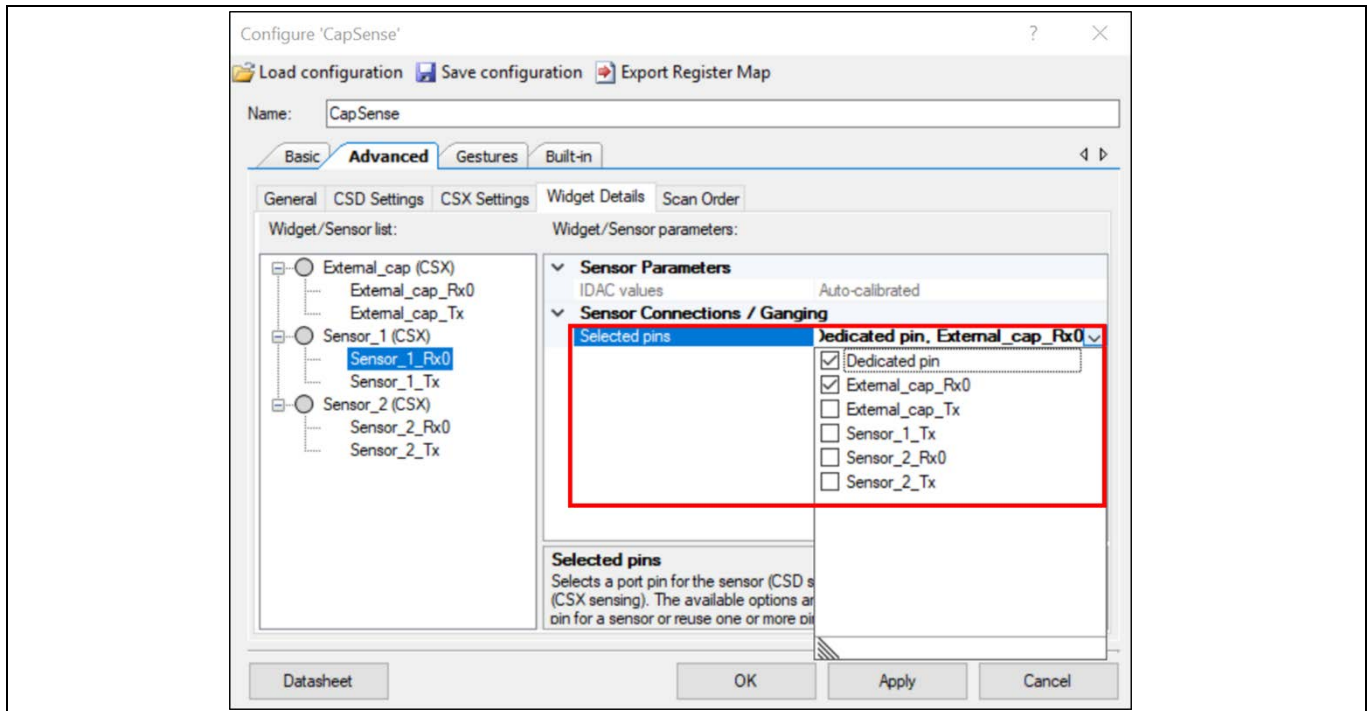


Figure 106 外部コンデンサとセンサーピンの取得

CAPSENSE™ のジェスチャー

6 CAPSENSE™ のジェスチャー

6.1 タッチジェスチャーの対応

PSoC™ 4 および PSoC™ 6 MCU の CAPSENSE™ コンポーネントは、スライダーおよびタッチパッドウィジェットのジェスチャー検出機能に対応します。スライダーとタッチパッドウィジェットのタッチパターンに基づいて、事前定義されたさまざまなジェスチャーを識別できます。

ジェスチャー検出機能は、選択したデバイスの製品番号で使用できることに注意してください。コンポーネントのジェスチャー機能を使用する場合は、この機能に対応するデバイスを選択していることを確認してください。

6.2 ジェスチャーグループ

ジェスチャーは、クリック、1本指スクロール、2本指スクロール、2本指ズーム、1本指エッジスワイプ、1本指フリック、および1本指回転、の複数のグループに分けられます。

Table 21 に、さまざまなウィジェットに対応するジェスチャーを示します。これらのジェスチャーの定義方法と、これらのジェスチャーを検出するために CAPSENSE™ コンフィギュレーターで設定する必要のあるパラメーターの詳細については、[コンポーネント データシート/ミドルウェア文書](#)を参照してください。

Table 21 さまざまな CAPSENSE™ ウィジェットに対応するジェスチャー

ウィジェットの種類	ジェスチャーグループ						
	クリック	1本指スクロール	2本指スクロール	1本指フリック	1本指エッジスワイプ	2本指ズーム	1本指回転
ボタン	✓	–	–	–	–	–	–
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	–	–	–	–	–	–	–
タッチパッド	✓	–	–	✓	–	–	✓
近接性	–	–	–	–	–	–	–

6.3 1本指ジェスチャーの実装

ジェスチャー検出の実装には、以下の手順が含まれます。

1. [ウィジェットの調整](#)
2. [事前定義されたジェスチャーの選択](#)
3. [タイムスタンプを使用したファームウェアの実装](#)
4. [ジェスチャーパラメーターの調整](#)

6.3.1 ウィジェットの調整

ウィジェットの CAPSENSE™ ハードウェアおよびソフトウェアパラメーターを調整してください。通常、ジェスチャーアプリケーションでは、指の動きの速度と方向が変化するため、指がウィジェットとほとんど接触しない場合があります。これは、ジェスチャーが実行されているときに [チューナー GUI](#) で

CAPSENSE™ のジェスチャー

重心データを表示することで確認できます。感度が十分であれば、途切れることなくデータを取得できます。重心データに途切れが見られる場合は、ジェスチャーのデータが完成し、途切れることなく表示されるまで感度を上げてください。

検出したい指のわずかな接触に対して、5:1 を超える SNR が得られることを確認してください。また、スライダーまたはタッチパッドの指の位置に対して線形の重心応答があることを確認してください。同じことを達成するため [スライダー ウィジェット調整](#) または [タッチパッド ウィジェット調整](#) セクションのガイドラインを使用してセンサーを調整してください。

6.3.2 事前定義されたジェスチャーの選択

まず、CAPSENSE™ コンポーネントの Gesture タブで **Gesture** を有効にする必要があります。ジェスチャーを有効にすると、ジェスチャーに関連するすべての設定パラメーターが表示されます。これらのパラメーターは、[Table 21](#) に示すように、ウィジェット/ジェスチャーグループによって体系的に配置されます。アプリケーションの要件に応じて、チェックボックスを選択することでジェスチャーを有効または無効にできます。ジェスチャーを有効にし、対応するパラメーターを設定するためには、次の手順を実行します。

- Widget ペインでジェスチャー機能を有効にする必要があるウィジェットを選択してください。プロジェクトに複数のウィジェットがある場合、PSoC™ Creator は 1 つのウィジェットのみでのジェスチャー認識を許可します。ただし、ModusToolbox™ では、複数のウィジェットでジェスチャー認識を有効にできません。
- Gesture ペインで目的のジェスチャーを選択してください。複数のジェスチャーを選択するオプションがあります。PSoC™ Creator では、スクロールジェスチャーとフリックジェスチャーを同時に有効にはできません。これは、スライダーとタッチパッドの両方に適用されます。ただし、ModusToolbox™ では、アプリケーションの要件に応じて複数のジェスチャーを有効にできます。
- Parameter ペインですべてのパラメーターを設定してください。ジェスチャーを選択すると、ウィンドウの右側のペインに、そのジェスチャーグループに関連するパラメーターが表示されます。[コンポーネント データシート/ミドルウェア文書](#) を参照してください。

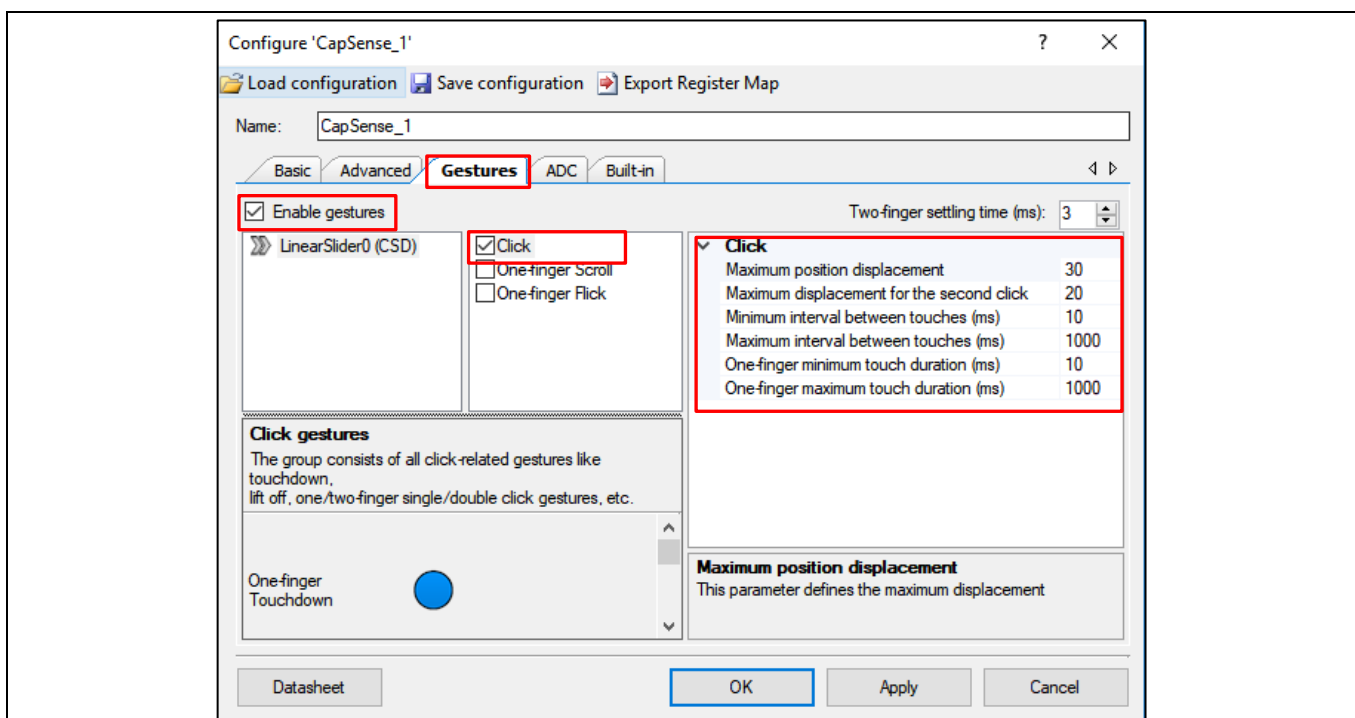


Figure 107 CAPSENSE™ コンポーネントのジェスチャー設定

CAPSENSE™ のジェスチャー

6.3.3 タイムスタンプを使用したファームウェアの実装

ジェスチャー認識のタイムスタンプを実装する方法を理解するためには、サンプルコード **PSoc™ 4 CAPSENSE™ Touchpad Gestures** を参照してください。各ジェスチャーには時間とともに変化するタッチのパターンがあるため、時間に関するタッチデータを適切に取得するためには、参照タイムスタンプが必要です。このタイムスタンプは、ジェスチャー認識アルゴリズムのサンプリングレートを表します。重心位置とそれぞれのタイムスタンプの両方がジェスチャーデコード API によって使用され、ウィジェットに適用可能なさまざまな事前定義されたジェスチャーパターンが決定されます。

まず、**ウィジェットの調整**で説明されている手順でウィジェットを調整し、ファームウェアでの2つの連続する CAPSENSE™ スキャン間の時間間隔を決定します。この期間でタイムスタンプを正確に更新します。それを正確に判断する方法は、CAPSENSE™ スキャンが完了した後にファームウェアの GPIO を切り替え、オシロスコープを使用して期間を見つけることです。

6.3.4 ジェスチャーパラメーターの調整

ここでは、スライダのジェスチャーパラメーターを設定する方法について説明します。同じ手順を、タッチパッドでサポートされているジェスチャーグループに拡張することもできます。CAPSENSE™ スライダは、クリック、1本指のスクロール、および1本指のフリックジェスチャー機能をサポートします。**コンポーネントデータシート/ミドルウェア文書**を参照してください。

6.3.4.1 ジェスチャーパラメーターを調整するためのチューナーGUIの使用

チューナーGUI の **Gesture View** を使用して、ジェスチャーパラメーターを調整し、エンドシステムで検出されたジェスチャーのパフォーマンスを視覚化および分析ができます。

ジェスチャーに**チューナーGUI**を使用するときは、次の点に注意してください。

1. 実行時にジェスチャーパラメーターを調整するためには、EZI2C で**チューナーGUI**を使用する必要があります。実行時に検出されたジェスチャーを視覚化するためには、同期通信モードを使用します。**チューナーGUI**の使用の詳細については、**コンポーネントデータシート/ミドルウェア文書**および**PSoc 4 CAPSENSE™ touchpad gestures code example**を参照してください。CAPSENSE™コンフィギュレーターで使用できるジェスチャーのすべてのパラメーターは**チューナーGUI**で使用でき、調整のためにこれらの値を直接編集できます。
2. **Figure 108**に示すように、Gesture View タブは以下のようにさまざまなペインに編成されます。

Gesture Event History ペインには、検出されたジェスチャーとウィジェット上のそれらの位置が表示されます。

Detected Gesture ペインは、検出されたジェスチャーを示します。遅延チェックボックスが有効になっている場合、指定された時間間隔でジェスチャー画像が表示されます。遅延が無効になっている場合、新しいジェスチャーが報告されるまで、最後に報告されたジェスチャー画像が表示されます。

チューナーGUIの **Cypress® Icon** は、スクロールジェスチャーに従って移動します。スクロールジェスチャーのパラメーターがどの程度適切に調整されているかを示します。この動的機能は、ジェスチャーパラメーターをさらに微調整するためのパフォーマンスフィードバックを提供します。

CAPSENSE™ のジェスチャー

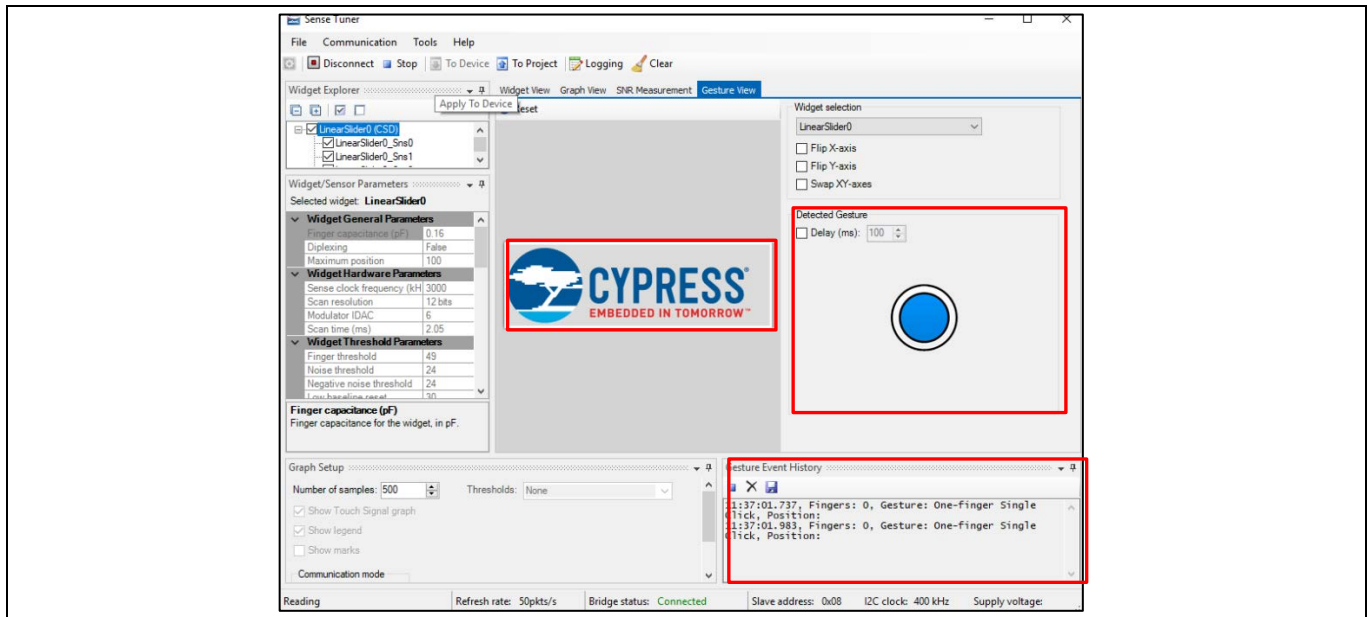


Figure 108 ジェスチャーのチューナーGUI

3. **チューナーGUI** を使用してイベント期間を決定します。イベント期間を決定する一般的な式は、**式 71** で与えられます。

式 71. ジェスチャーの持続時間

$$\text{イベント期間} = \text{サンプルの数} \times T_{\text{sample}}$$

ここで、

サンプルの数 = ジェスチャーイベントが発生したサンプルの数です。このデータは、**チューナーGUI** のグラフビューから取得できます。

T_{sample} = 2 つのサンプル間の時間間隔です。

$$T_{\text{sample}} = \frac{1}{\text{Refresh rate}}$$

CAPSENSE™ のジェスチャー

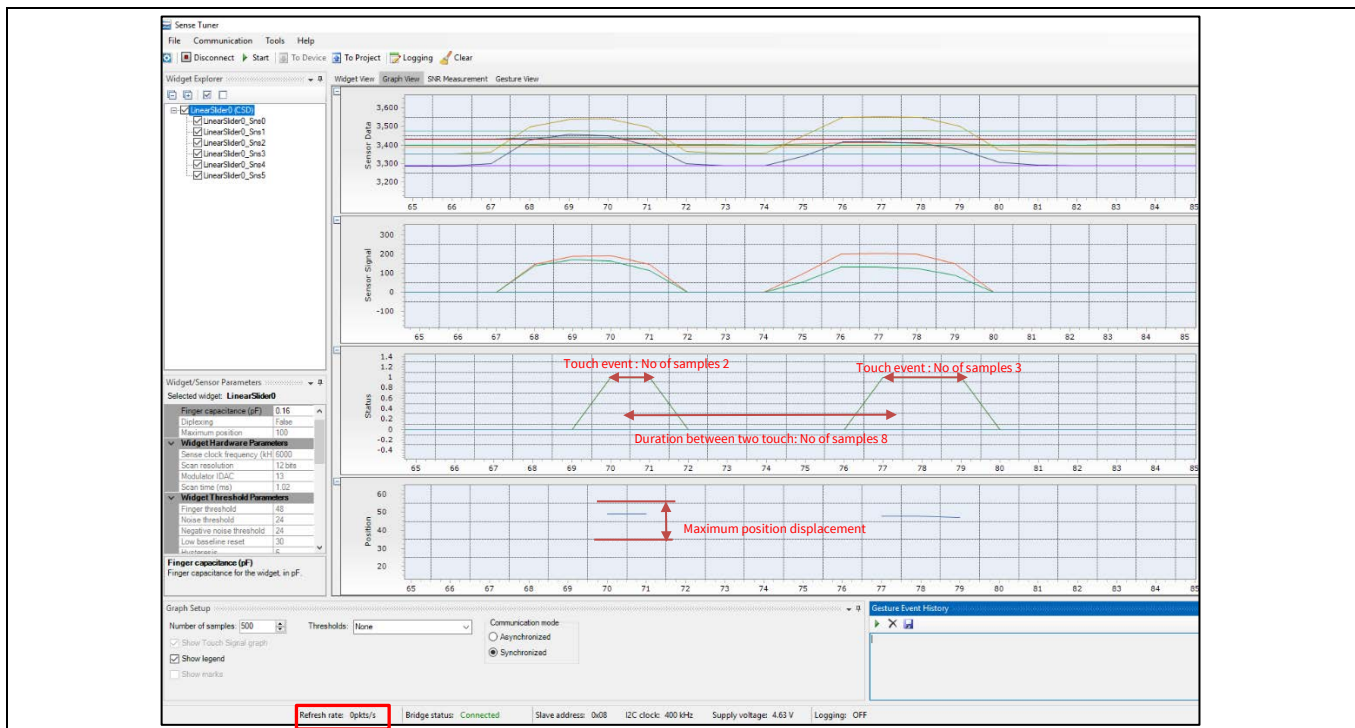


Figure 109 チューナーGUIを使用したジェスチャーパラメーターの決定

6.3.4.2 クリック

クリックジェスチャーには、シングルクリックとダブルクリックの2種類があります。Table 22 に、PSoC™ Creator と ModusToolbox™ の両方でクリックジェスチャー用に設定する必要があるパラメーターの一覧を示します。コンポーネントデータシート/ミドルウェア文書を参照してください。Table 23 に、クリックジェスチャーのジェスチャーパラメーターの推奨値を示します。

Table 22 クリックジェスチャーパラメーター

ジェスチャー	PSoC™ Creator	ModusToolbox™
シングルクリック	1 本指の最小タッチ時間	最小クリックタイムアウト
	1 本指の最大タッチ時間	最大クリックタイムアウト
	最大位置変位	最大クリック距離
ダブルクリック	タッチ間の最小間隔	2 回目のクリック最小間隔
	タッチ間の最大間隔	2 回目のクリック最大間隔
	2 回目のクリックの最大変位	2 回目のクリック最大距離

Table 23 クリックジェスチャーの推奨値

パラメーター	標準値
最大位置変位	スライダーの最大位置の 20%
2 回目のクリックでの最大位置変位	スライダーの最大位置の 20%
タッチ間の最小間隔 (ms)	60
タッチ間の最大間隔 (ms)	400
1 本指の最小タッチ時間 (ms)	20
1 本指の最大タッチ時間 (ms)	400

CAPSENSE™ のジェスチャー

6.3.4.2.1 シングルクリック

シングルクリックは、タッチダウンイベントとそれに続くリフトオフとして定義されます。**Figure 110** に、有効なシングルクリックイベントのために満たす必要のある空間条件とタイミング条件を示します。

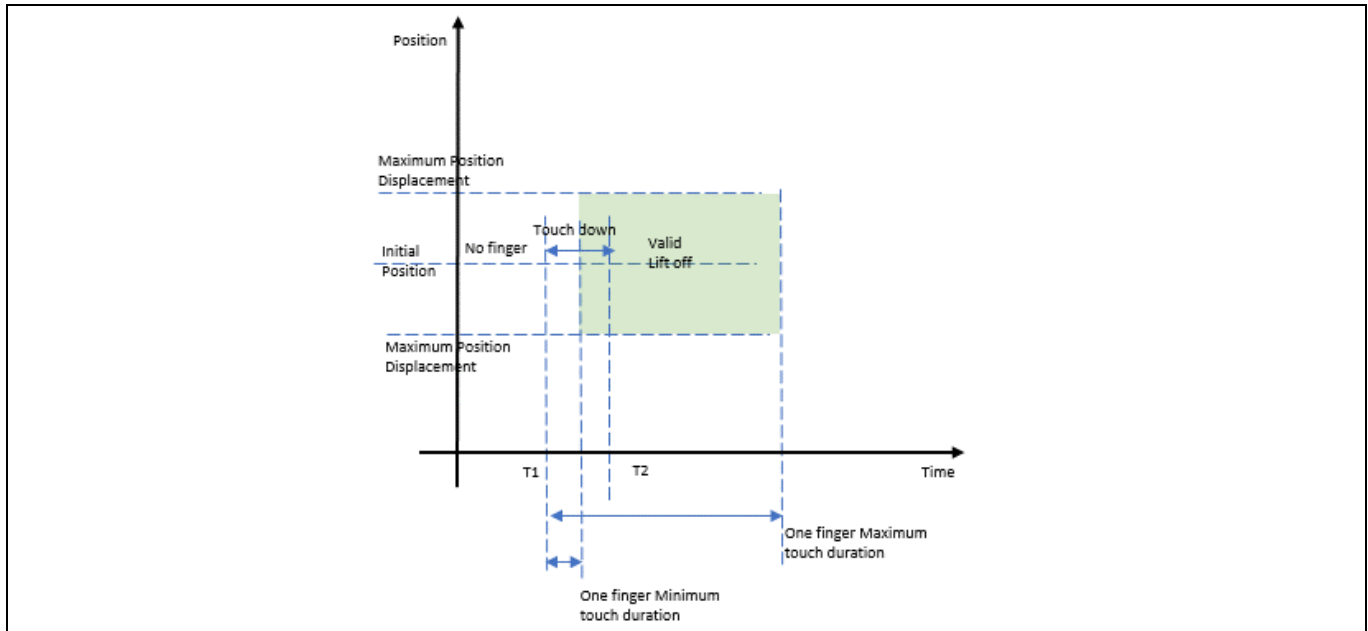


Figure 110 シングルクリックジェスチャー

Figure 110 から、時間 T1 で、指がスライダーに着地しました。時間 T2 で、指がスライダーから持ち上げられます。有効なシングルクリックの場合、タッチダウン時間は”One finger Minimum touch duration”と”One finger Maximum touch duration”の間にあり、タッチの初期位置からのリフトオフの相対位置は”Maximum Position Displacement”パラメーターよりも小さい必要があります。

各シングルクリックイベントの期間は、**Figure 109** に示すように、**チューナーGUI** の **Graph view** でシングルクリックのサンプル数とリフレッシュレートを見つけることにより、**式 71** を使用して決定できます。シングルクリックイベント期間から、パラメーター”One finger Minimum touch duration”と”One finger Maximum touch duration”を修正します。Maximum Position Displacement パラメーターは、**Figure 109** に示すように、**チューナーGUI** を使用して重心位置の最大変動を観察することで決定できます。**Table 23** に示すように、推奨値はスライダーの最大重心位置の 20 パーセントです。

6.3.4.2.2 ダブルクリック

ダブルクリックとは、2 回のシングルクリックイベントが次々に発生し、2 回目のクリックが 2 回のタッチ間の最小時間間隔と最大時間間隔の間に発生することです。さらに、タッチダウンイベントの初期位置からの 2 回目のクリックの相対位置は、2 回目のクリックの最大位置変位よりも小さくする必要があります。**Figure 111** に、有効なダブルクリックイベントのために満たす必要のある空間条件とタイミング条件を示します。

CAPSENSE™ のジェスチャー

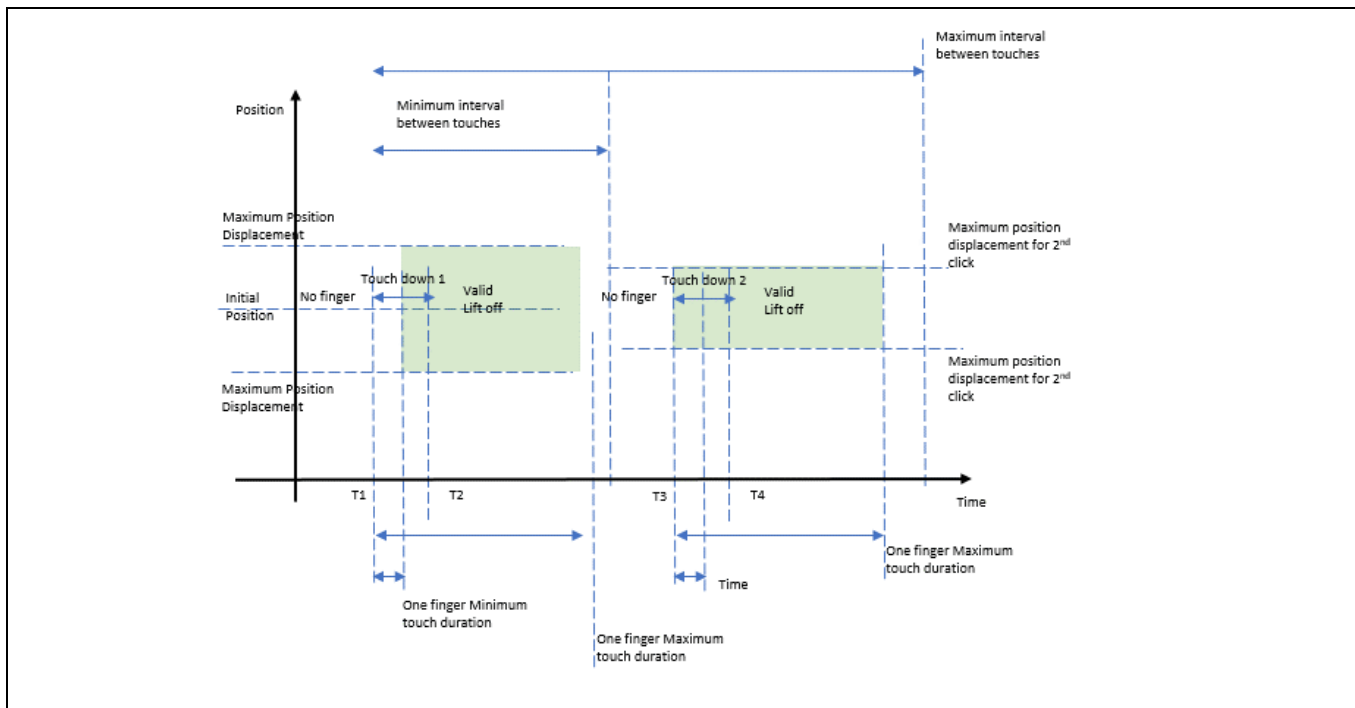


Figure 111 ダブルクリックジェスチャー

Figure 111 から、時間 T1 で最初のクリックで指がスライダーにタッチダウンし、時間 T2 で指がスライダーから持ち上げられます。T3 で 2 回目のクリックで指がスライダーをタッチダウンし、T4 で指がスライダーから持ち上げられます。有効なダブルクリックの場合、各クリックはシングルクリックの条件を満たす必要があります、2 回目のクリックはタッチパラメーター間の最小間隔と最大間隔の間に発生する必要があります。

チューナー GUI の Graph view を使用して、ダブルクリックのタッチデータを観察します。シングルクリックで説明したように、シングルクリックのパラメーターを決定します。これらの後で、チューナー GUI の Graph view を使用して 2 つのタッチ間の期間を決定し、タッチパラメーター間の最小間隔と最大間隔の値を設定します。ダブルクリックイベントでキャプチャされた一般的なデータを Figure 109 に示します。

6.3.4.3 スクロール

スライダーで検出できる 2 つの異なるスクロールジェスチャーがあります。1 本指のスクロールと 1 本指の慣性スクロールです。コンポーネントデータシート/モジュールウェア文書を参照してください。

Table 24 に、スクロールジェスチャー用に設定する必要があるパラメーターを示します。ModusToolbox™ では、1 本の指の慣性スクロールジェスチャーには対応しないことに注意してください。

Table 24 クリックジェスチャーパラメーター

ジェスチャー	PSoC™ Creator	ModusToolbox™
1 本指のスクロール	位置しきい値 N	最小スクロール距離
	スクロールステップ	-
	デバウンス	スクロール デバウンス
1 本指の慣性スクロール	位置慣性しきい値	該当なし
	カウントレベル	

CAPSENSE™ のジェスチャー

6.3.4.3.1 1 本指のスクロール

1 本指でスクロールするジェスチャーは、タッチダウンとそれに続く特定の方向への移動の組合せです。2つの連続するスキャン間の位置の変化は、チューニング後にコンフィギュレーターで指定された位置しきい値を超える必要があります。[コンポーネント データシート/ミドルウェア 文書](#)を参照してください。

Table 24 に示すように、以下を実行してスクロールジェスチャーパラメーター値を設定してください。

1. **チューナー GUI** の **Graph View** (重心位置) からスクロールジェスチャーのサンプル数を決定してください。
2. **式 71** を使用して、完全なスクロールの期間を決定してください。
3. **チューナー GUI** を使用して、完全なスクロールの図心位置の変化を決定してください。
4. **式 72** から位置しきい値を決定してください。各ジェスチャーは、アプリケーションコードのタイムスタンプに設定されているサンプルレートでスキャンされます。位置のしきい値は、タイムスタンプに設定されている期間の重心位置の変化によって与えられます。

式 72. 位置しきい値を決定する式

$$\text{位置しきい値} = \frac{\text{重心位置での変化}}{\text{総スクロール期間}} \times \text{タイムスタンプ期間}$$

5. PSoC™ Creator で、4つの異なる位置しきい値とそれらのスクロールカウント値をコンフィギュレーターで設定します。これらは、スクロールジェスチャーの速度を変えることによって決定されます。ここで、スクロールの速度を変更し、手順 1~4 を繰り返して、これらの位置しきい値を設定します。ModusToolbox™には、パラメーターが1つだけあります。最小スクロール距離。位置しきい値を決定したのと同じ方法でその値を決定します。
6. CAPSENSE™ データ構造からスクロールステップを読み出し、それを使用してスクロールジェスチャーの速度と滑らかさを制御します。スクロールステップは、位置しきい値によって異なります。このスクロールステップは、スクロールに関して変更される実際の変数値を制御するためにアプリケーションコードで使用されます。スクロールステップパラメーターは ModusToolbox™ では使用できないことに注意してください。
7. 原則として、スライダの最大位置をスライダの寸法の 10 倍に設定します。scrollDistanceMin = 10 を設定すると、1mm 未満の動きはすべてスクロールジェスチャーを検出しません。この数を超えるものはすべて、ジェスチャーを検出する可能性があります。

チューナー GUI (**Figure 108**) の **Cypress®** アイコンを観察して、特定のハードウェアでのスクロールジェスチャーのチューニングがどの程度適切に行われたかについてのフィードバックを取得してください。UART をスクロールして制御する必要のある変数を印刷して、スクロールに対して値がどのように変化しているかも視覚化できます。これは視覚的なフィードバックとして使用できます。スクロールに関する変数値の変動がエンドユーザーアプリケーションの要件を満たすまで、位置しきい値パラメーターと対応するステップ数を調整する必要があります。

6.3.4.3.2 1 本指の慣性スクロール

1 本指の慣性スクロールジェスチャーは、タッチダウンイベントと、それに続く特定の方向への最小変位、そしてリフトオフとして定義されます。スクロールの移動は、変数の終了値に達すると自動的に停止します。[コンポーネント データシート/ミドルウェア 文書](#)を参照してください。

ジェスチャーパラメーターを **Table 24** に示します。位置慣性しきい値パラメーターは、リフトオフの前に必要な重心位置の最小変化によって与えられます。その値は、**1 本指のスクロール**の手順で決定する

CAPSENSE™ のジェスチャー

こともできます。カウント値パラメーターは、スクロールの勢いを定義します。それは2つの可能な値(低または高)を取れます。最終アプリケーションの要件に応じてカウント値を選択してください。

6.3.4.4 1本指のフリック

フリックジェスチャーは、タッチダウンイベントと、それに続く高速変位およびリフトオフイベントです(コンポーネントデータシート/ミドルウェア文書を参照)。フリックジェスチャーは、1本指慣性スクロールに似ています。唯一の違いは、コンフィギュレーターで定義された最大サンプル間隔内で、高速変位とそれに続くリフトオフイベントが必要なことです。1本指のスクロールと同じ手順に従い、式71を使用して、位置しきい値と最大サンプル間隔を決定できます。

Table 25 1本の指のフリックジェスチャーパラメーター

ジェスチャー	PSoC™ Creator	ModusToolbox™
1本指のフリックジェスチャー	位置しきい値	最小フリック距離
	最大サンプル間隔	最大フリックタイムアウト

6.4 2本指のジェスチャーの実装

タッチパッドウィジェットでは、2本指のスクロールや2本指のズームなどの2本指のジェスチャーに対応します。この機能は、タッチパッドウィジェットのWidget Detailタブで有効にする必要があります。パラメーターを調整する手順は、1本指ジェスチャーの実装で説明した手順と同じです(コンポーネントデータシート/ミドルウェア文書を参照)。Figure 112に、コンフィギュレーターで2本指のタッチジェスチャーを有効にし、Centroid typeを5x5 Centroidとして選択し、Two finger detectionをTrueに設定する方法を示します。

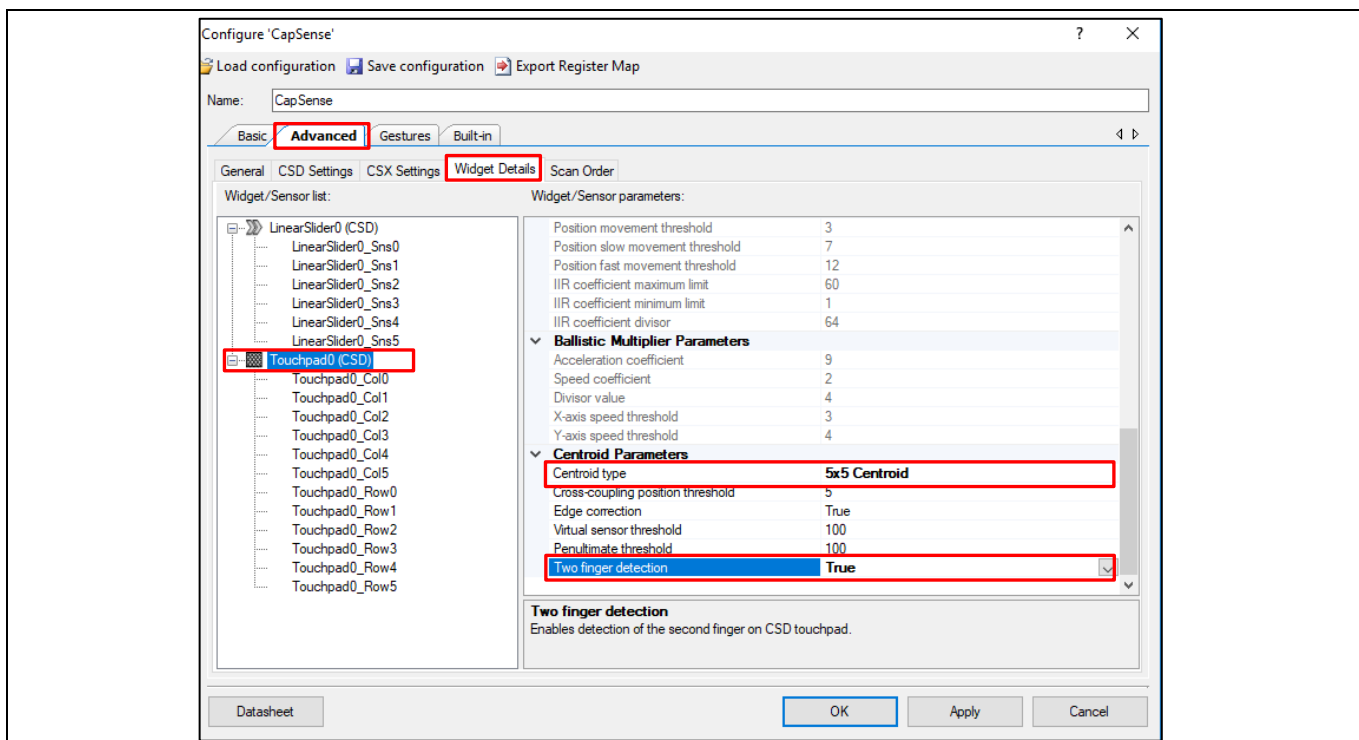


Figure 112 CAPSENSE™コンポーネントで2本指のタッチジェスチャーの有効化

CAPSENSE™ のジェスチャー

6.5 ジェスチャーの高度なフィルター

弾道乗算器やアダプティブ IIR フィルターなどのジェスチャー用のより高度なフィルタリング機能と、ジェスチャー認識とユーザーエクスペリエンスを向上させるエッジ補正機能を提供します。

[コンポーネント データシート/ミドルウェア文書](#)を参照してください。

設計上の注意事項

7 設計上の注意事項

本章では、CAPSENSE™のファームウェアとハードウェアの設計上の注意事項について説明します。

7.1 ファームウェア

CAPSENSE™コンポーネントは、ファームウェア開発を簡素化するために、複数のアプリケーションプログラミングインターフェース (API) を提供します。CAPSENSE™**コンポーネント データシート**は使用可能な API の詳細リストと解説を提供します。回路図入力とファームウェア開発を学ぶために、PSoC™ Creator または ModusToolbox™で提供される CAPSENSE™の**サンプルプロジェクト**が使用できます。詳細は**第4章**を参照してください。

CAPSENSE™スキャンは本質的にノンブロッキングです。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。そのため、CAPSENSE™スキャンが進行中、他のタスク実行に CPU を使用できません。しかし、CAPSENSE™は高感度アナログシステムであることに注意してください。したがって、デバイス電流の急激な変化は raw カウントに乗るノイズを増加させることがあります。近接センサーか厚いオーバーレイ付きのボタンなどの高感度を必要とするウィジェットを使っている場合は、ブロッキングスキャンを使用する必要があります。ノンブロッキングスキャンのファームウェア例は以下のとおりです。

Code Listing 1

```
/* Enable global interrupts */
CyGlobalIntEnable;

/* Start EZI2C component */
EZI2C_Start();

/*
 * Set up communication data buffer to CapSense data structure to be
 * exposed to I2C master at primary slave address request.
 */
EZI2C_EzI2CSetBuffer1(sizeof(CapSense_dsRam),
sizeof(CapSense_dsRam),
(uint8 *)&CapSense_dsRam);

/* Initialize CapSense component */
CapSense_Start();
/* Scan all widgets */
CapSense_ScanAllWidgets();

for(;;)
{
    /* Do this only when a scan is done */
    if(CapSense_NOT_BUSY == CapSense_IsBusy())
    { /* Process all widgets */
        CapSense_ProcessAllWidgets();
        /* Scan result verification */
        if (CapSense_IsAnyWidgetActive())
        {
            /* Add any required functionality
            based on scanning result */

```


設計上の注意事項

Code Listing 1

```
    }
    /* Include Tuner */
    CapSense_RunTuner();
    /* Start next scan */
    CapSense_ScanAllWidgets();
}
/* CPU Sleep */
CySysPmSleep();
}
```

高感度 CAPSENSE™ スキャンが進行中では、割込みコード、電力モードの遷移およびペリフェラルのオン/オフの切り替えを回避する必要があります。しかし、高感度ウィジェットを使用していない場合、他のタスクを実行するために CPU を使用できます。次の節で説明されるように、CAPSENSE™ システムの平均消費電力を下げるために、PSoC™ 4 の低消費電力モードも使用できます。ノンブロッキングコードを使っている場合、チューナー GUI を使用し、raw カウントおよび SNR を監視して検証することを推奨します。

ModusToolbox™ ソフトウェアを使用してファームウェアを開発する場合、本文書の [ModusToolbox™ セクション](#) を参照してください。

7.1.1 低消費電力の設計

PSoC™ の低消費電力モードにより、本質的な機能性を保持しながら、全体の電力消費を削減できます。PSoC™ 4 低消費電力モードの基礎知識は [AN86233 - PSoC™ 4 low-power modes and power reduction techniques](#) を、PSoC™ 6 の低電力モードについては、[AN219528 - PSoC™ 6 MCU low-power modes and power reduction techniques](#) を、低消費電力 CAPSENSE™ アプリケーションの設計については [AN210998 - PSoC™ 4 low-power CAPSENSE™ design](#) を参照してください。

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。ファームウェアは、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できます。CSD ハードウェアのスキャンが完了した後、割込みを生成し、デバイスはアクティブモードに復帰させます。

CAPSENSE™ システムの消費電力を削減するためのさまざまなファームウェアおよびハードウェア技術があります。

- 複数のウィジェットを一緒にスキャンする API を使用する場合、デバイスは 1 つのウィジェットのスキャンを終了した後にアクティブモードに戻ります。したがって、デザインの消費電力を削減するために、各ウィジェットを個別にスキャンする必要があります。CAPSENSE™ コンポーネントのデータシートを参照してください。
- PSoC™ のディープスリープモードを使用して、CAPSENSE™ デザインの消費電力を大幅に削減できます。ただし、CAPSENSE™ ハードウェアはディープスリープモードでは無効になります。したがって、タッチをスキャンするためには、デバイスを頻繁にウェイクアップする必要があります。PSoC™ のウォッチドッグタイマー (WDT) を使用して、デバイスをディープスリープモードから頻繁にウェイクアップできます。スキャンの頻度を増やすと、CAPSENSE™ システムの応答が向上しますが、平均消費電力も増加します。
- デザイン内のセンサーの数が増えると、デバイスはすべてのセンサーをスキャンするためにアクティブモードでより多くの時間を費やす必要があります。これにより、平均消費電力が増加します。複数のセンサーを備えた設計で電力を節約するためには、すべてのセンサーを囲む個別の [近接ループ](#) を含める必要があります。デバイスがディープスリープモードからウェイクアップするときは、

設計上の注意事項

この近接センサーのみをスキャンしてください。近接センサーがアクティブな場合、デバイスはアクティブモードのまま、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の場合、デバイスはディープスリープモードに戻れます。Figure 113 に、このプロセスを示します。

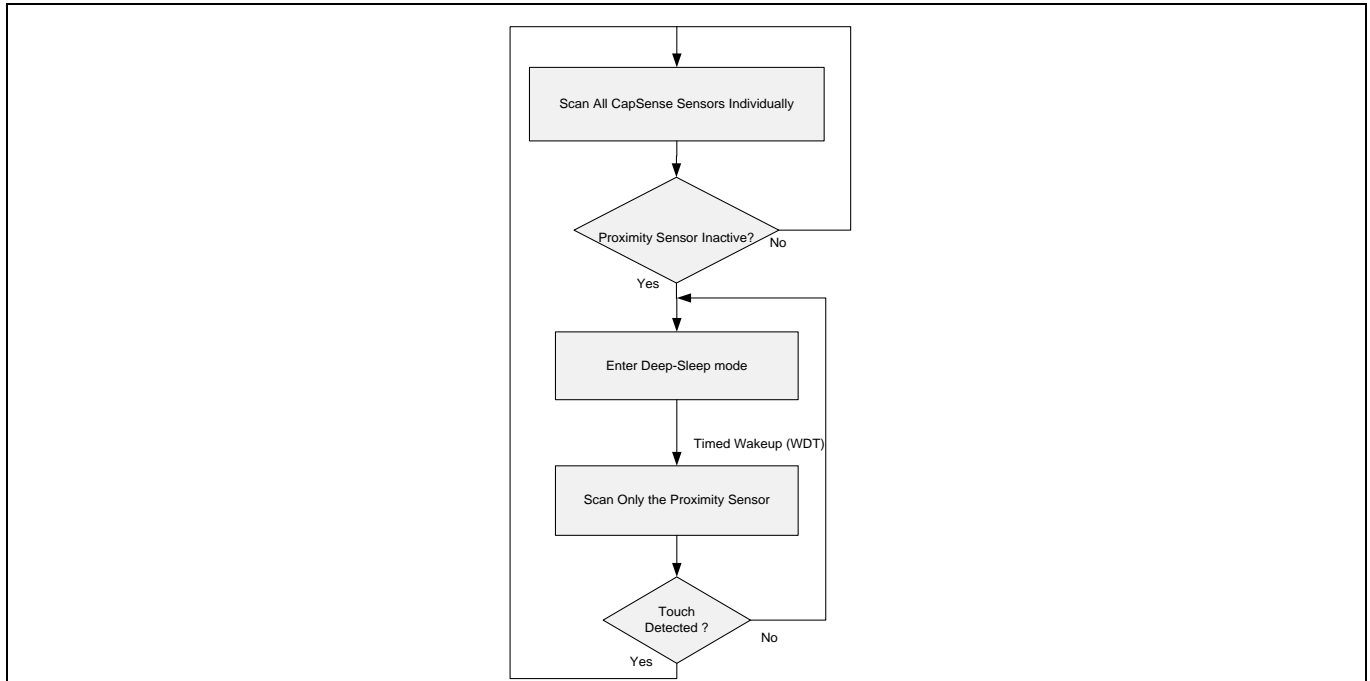


Figure 113 低消費電力 CAPSENSE™ の設計

- CAPSENSE™ コンポーネントは、センサーを連動させ、アプリケーションレベルでスキャンを管理することにより、スキャンの実行時間を短縮することにより、消費電力を削減できます。この場合、設計内のすべてのセンサーは「ganged (連動)」します。つまり、AMUX バスに同時に接続して、仮想センサーを形成します。連動センサーの実装の詳細については、サンプルコード [PSoC™ 4 low power ganged sensor](#) および [AN92239 - Proximity sensing with CAPSENSE™](#) を参照してください。連動センサーは、センサーを個別に検討する場合と比較してプロパティが異なるため、調整パラメーターが異なります。したがって、単一の CSD ボタンと見なし、個別に調整する必要があります。[手動チューニング](#) を参照してください。連動センサーは、ウォッチドッグタイマー (WDT) を使用して定期的にスキャンされます。連動センサーがタッチイベントを報告した場合は、スキャンする必要のある実際のウィジェットのスキャンを有効にします。これは、Wake on Touch モードを必要とする CAPSENSE™ デザインで役立ちます。手順は、Figure 113 で説明した手順と同様です。CAPSENSE™ とウェイクアップ間隔を適切に調整することにより、良好なタッチ応答を維持しながら、非常に低いシステム電流を実現できます。この手法は、CSX タッチパッドウィジェットでも使用できます。
- システムタイマーや I²C などの高速周辺機器が必要な場合は、ディープスリープモードにする代わりに CPU をスリープモードにできます。
- [被駆動シールド信号およびシールド電極](#) で説明されているように、デザインにシールドハッチを追加して、寄生容量を減らし、スキャン時間を短縮もできます。スキャン時間と消費電力は直接関係します。したがって、スキャン時間を短縮することにより、消費電力が削減されます。

Note: PSoC™ 4000 デバイスでは、CAPSENSE™ スキャンが進行中の場合にスリープモードに入ることは推奨しません。

設計上の注意事項

7.2 センサーの構造

静電容量センサーは、アプリケーションの要件に応じて、異なる材料を使用して設定できます。典型的なセンサー設定では、タッチを検知する導電性パッドまたは表面が導電性配線またはリンクにより PSoC™ のピンに接続されます。この全体の構成は非導電性オーバーレイ材料の下に配置され、ユーザーはオーバーレイの上で操作します。

Figure 114 に、最も一般的な CAPSENSE™ センサーの構造を示します。

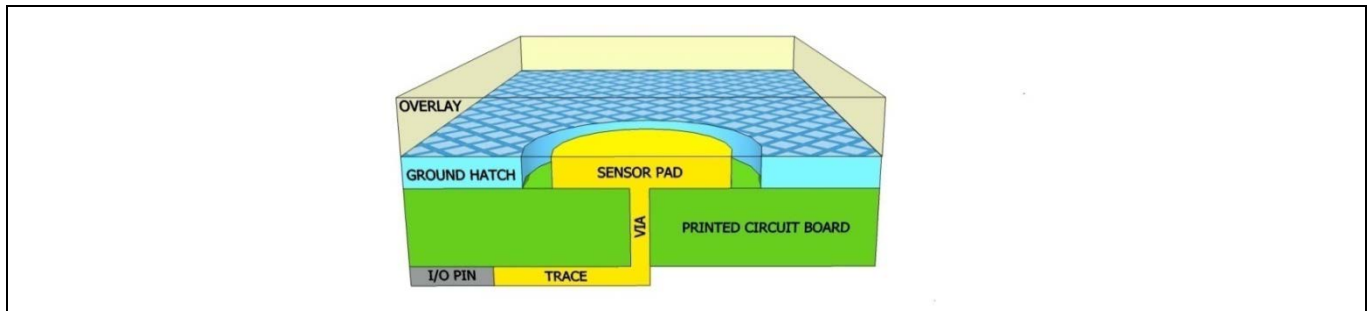


Figure 114 CAPSENSE™ センサーの構造

PCB の表面にエッチングされた銅パッドは CAPSENSE™ センサーとして機能します。非導電性のオーバーレイ素材が、タッチ表面として機能します。オーバーレイにより、センサーを環境から保護し、指の直接接触が回避できます。センサーパッドを取り囲んでいるグランドハッチは、センサーを他のセンサーおよび PCB 配線から分離します。

耐液性が要求されると、グランドハッチの代わりにシールドハッチを使用する必要があります。この場合、ハッチをグランドに接続する代わりに、シールド信号で駆動します。詳細は[耐液性](#)を参照してください。

最も基本的な CAPSENSE™ PCB 形状は 2 層基板で、上面にセンサーパッドとハッチンググランド面があり、下面に電気部品が搭載されます。Figure 115 に CAPSENSE™ ハードウェアの分解図を示します。

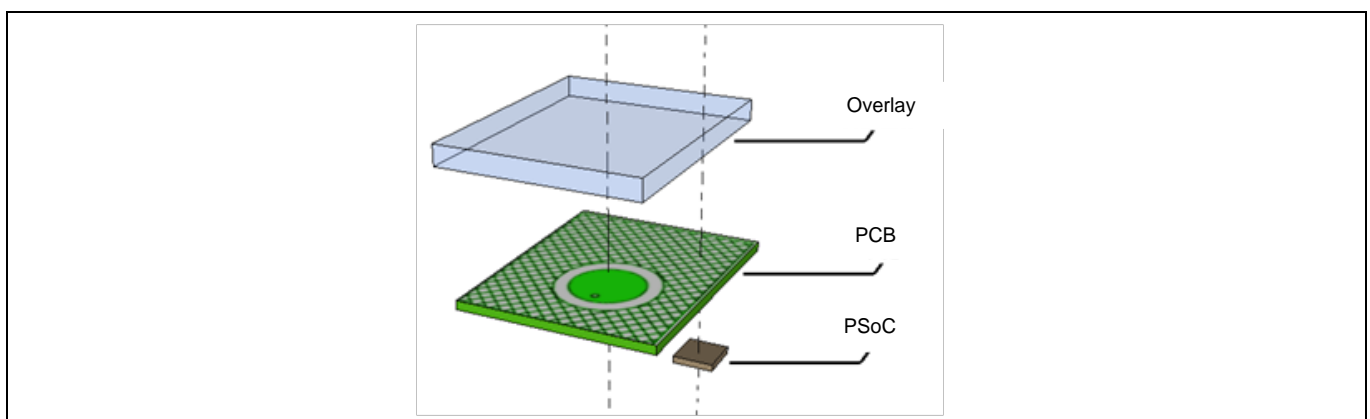


Figure 115 CAPSENSE™ ハードウェア

センサーはインジウムスズ酸化物 (ITO) またはガラスや柔軟なプリント基板の上に印刷されたインクなど銅以外の材料を用いて構築できます。Figure 106 に示すように、スプリングはオーバーレイが PCB から離れた距離に配置することを可能にするスプリングセンサーを作成するために、CAPSENSE™ センサーとして使用される場合があります。スプリングセンサーおよび ITO や印刷済みインクなど他の非銅センサーの PCB 設計上の注意事項は [Getting started with CAPSENSE™ design guide](#) を参照してください。

設計上の注意事項

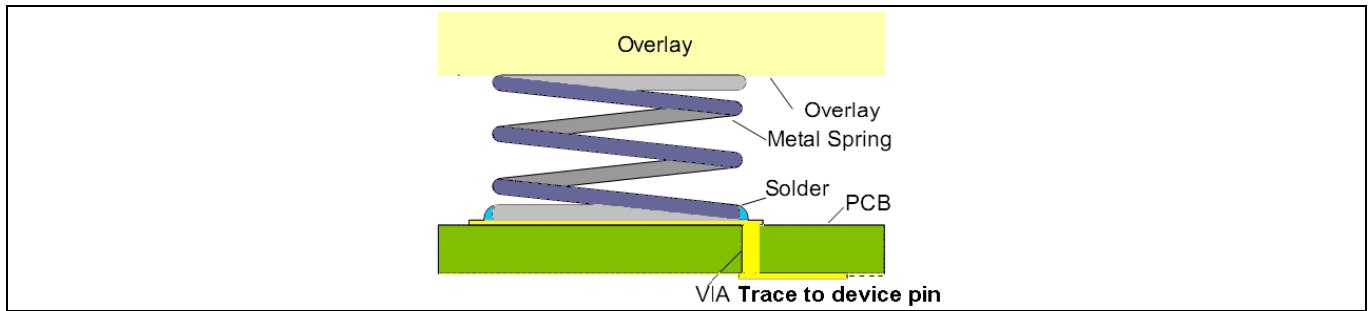


Figure 116 センサーとしてスプリングを使用するセンサーの構造

7.3 オーバーレイの選択

7.3.1 オーバーレイの素材

このオーバーレイは、指の静電容量の大きさを決めるため、CAPSENSE™ハードウェアの重要な部分です。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ます。詳細は[指の静電容量](#)を参照してください。

Table 26 にいくつかのオーバーレイ素材の比誘電率を示します。比誘電率 2.0~8.0 の素材は、CAPSENSE™オーバーレイに適します。

Table 26 オーバーレイ素材の比誘電率

素材	ϵ_r
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 (Plexiglas®)	2.8
ABS	2.4~4.1
木造のテーブルおよびデスクトップ	1.2~2.5
ジブサム (乾式壁)	2.5~6.0

Note: 導電性材料は、電界パターンを妨げます。そのため、オーバーレイに導電性材料を使用してはいけません。オーバーレイに導電性塗料も使用しないでください。

7.3.2 オーバーレイの厚さ

指の静電容量は、オーバーレイの厚さに反比例します。そのため、薄いオーバーレイは厚いオーバーレイより多くの信号を与えます。詳細は[指の静電容量](#)を参照してください。

Table 27 にさまざまな CAPSENSE™ウィジェット用のアクリル オーバーレイの推奨される最大の厚さを示します。

Table 27 アクリル オーバーレイの最大の厚さ

設計上の注意事項

ウィジェット	最大の厚さ (mm) - 第 4 世代 CAPSENSE™	最大の厚さ (mm) - 第 5 世代 CAPSENSE™
Button	5	18
Slider	5 ¹⁰	18
Touchpad	0.5	3

指の静電容量もオーバーレイの比誘電率に依存するため、比誘電率もオーバーレイの最大厚さのガイドラインで重要な役割を果たします。通常ガラスの比誘電率は約 ($\epsilon_r=8$) ですが、アクリルの比誘電率は約 ($\epsilon_r=2.5$) です。 ($\epsilon_r/2.5$) の比率は同じ感度レベルでのプラスチックに対するオーバーレイの厚さの推定比です。この経験則を使用すると、通常ガラスオーバーレイの厚さはプラスチックオーバーレイの厚さの3倍ですが、同じ感度レベルを維持できます。

さらに、非常に薄いオーバーレイやオーバーレイを使用しないでください。次の理由により、CAPSENSE™ デザインではオーバーレイの厚さを最小にすることが重要です。

- オーバーレイは、環境条件からの保護を提供し、指が直接接触するのを防ぎ、ESD 保護を提供します。オーバーレイの厚さは、良好な信号を提供するために十分に薄くする必要があります。ボタンのサイズと ESD に耐える強度に基づいて決定されます。 [AN64846 Getting started with the CAPSENSE™](#) を参照してください。
- CSD ボタンの場合、オーバーレイがない場合、ボタンは過敏になります。
- スライダの場合、オーバーレイがない場合、スライダーセグメントの生のカウントが飽和し、スライダーの重心応答が非線形になる可能性があります。 [スライダーのデザイン](#) を参照してください。
- CSX センサーの場合、オーバーレイの最小厚さを 0.5 mm にすることを推奨します。違反した場合、センサーに指を置いたり、Tx 電極と Rx 電極に水滴が落ちたりすると、raw カウントの急激な減少が見られます。 [接地の影響](#) を参照してください。

7.3.3 オーバーレイの接着剤

オーバーレイは、PCB と良好な機械的な接触を維持しなければなりません。オーバーレイを PCB に接着するためには非導電の接着剤フィルムを使用する必要があります。このフィルムは、オーバーレイとセンサーパッド間の空隙を無くすことによってシステムの感度を増加させます。3M™ は、CAPSENSE™ アプリケーションに広く使用される 200MP と呼ばれる高性能のアクリル系接着剤を提供します。これは、接着剤転写テープ形状で使用されます。製品例の番号は 467MP と 468MP です。

7.4 PCB レイアウトガイドライン

PCB レイアウトガイドラインは、良好な感度と高い [信号対ノイズ比](#) を備えた CAPSENSE™ システムを設計するために役立ちます。

7.4.1 センサー C_p

CAPSENSE™ システム設計では、センサーの C_p はデバイスのサポートされている範囲内にある必要があります。サポートされている C_p 範囲は、[コンポーネント データシート/ミドルウェア文書](#) にあります。 C_p の主なコンポーネントは、デバイスのトレース容量、センサーパッド容量、およびピン容量です。ピン容量はデバイスに依存するため ([デバイスデータシート](#) を参照)、データシートの C_p 基準を満たせるようにセンサーとトレース容量のみを設計できます。 C_p と PCB レイアウト機能の関係は単純ではありません。

¹⁰ 5mm のアクリルオーバーレイでは、SmartSense コンポーネントはスライダー動作のために、最小 9mm の指の直径を必要とします。指の直径が 9mm 未満の場合、手動チューニングを使用する必要があります。

設計上の注意事項

ません。C_pは、センサーパッドのサイズとトレースの長さや幅が大きくなると、センサーパッドとグラウンドハッチの間のギャップが小さくなると大きくなります。

C_pを下げる方法はたくさんあります。

- トレースの長さや幅をできるだけ小さくしてください。トレース長を短くすると、ノイズ耐性が向上します。
- シールド信号でハッチを駆動してください。[被駆動シールド信号およびシールド電極](#)を参照してください。

センサーパッドのサイズを小さくすると、指の静電容量も減少するため、推奨しません。センサーパッドが小さく、センサーパッドがデバイスの近くに配置されているためにトレース長が非常に短いなどの特殊なケースでは、センサーのC_pがデバイスでサポートされている最小C_pよりも低くなる可能性があります。その場合、センサーまたは未使用のピンの両端にコンデンサのフットプリントを追加します。C_pがサポートされる範囲よりも低いことが確認された場合は、センサーの両端または未使用のピンに4.7 pFのコンデンサを配置し、CAPSENSE™ スキャン中にコンデンサを連動します。詳細についてはFAQ [5.3.7.7](#)を参照してください。これにより、センサーのC_pがサポートされる範囲まで増加します。

トレースが長い場合、または地面が近くにあるためにセンサーのC_pが非常に高い場合は、C_p値が高いために感度が低下しないように、相互容量センシング方式を使用してください。相互容量センシング方式におけるCAPSENSE™ センサーの感度は、センサーC_pとは無関係です。

7.4.2 回路基板の層

ほとんどのアプリケーションは、センサーパッドとハッチンググラウンド面が上面に、他のすべての部品が下面にある2層基板を使用しています。もっと複雑なPCBは4層を使用します。

- FR4 ベースの PCB 設計は、回路基板の厚さが 0.020 インチ (0.5mm) から 0.063 インチ (1.6mm) の間で正常に動作します。
- フレックス回路は CAPSENSE™ でも良好に作用します。曲面にはフレックス回路を使用できます。本書のすべての PCB ガイドラインもフレックス回路に適用します。CAPSENSE™ に厚さが 0.01 インチ (0.25mm) 以上のフレックス回路を使用する必要があります。フレックス回路に使用される Kapton® 素材の高い絶縁破壊電圧 (290kV/mm) は、CAPSENSE™ センサーへの内蔵の ESD 保護を提供します。

7.4.3 ボタンデザイン

7.4.3.1 自己容量のボタンデザイン

自己容量ボタンには単一の電極があり、以下に推奨するようにさまざまな形状とサイズにできます。

形状 円形のセンサーパッドを CAPSENSE™ ボタンに使用する必要があります。角を丸くした長方形も良好です。しかし、尖った角 (<90°) は電場を集中するため、このような角を避けてください。[Figure 117](#) に、推奨されるボタン形状を示します。

設計上の注意事項

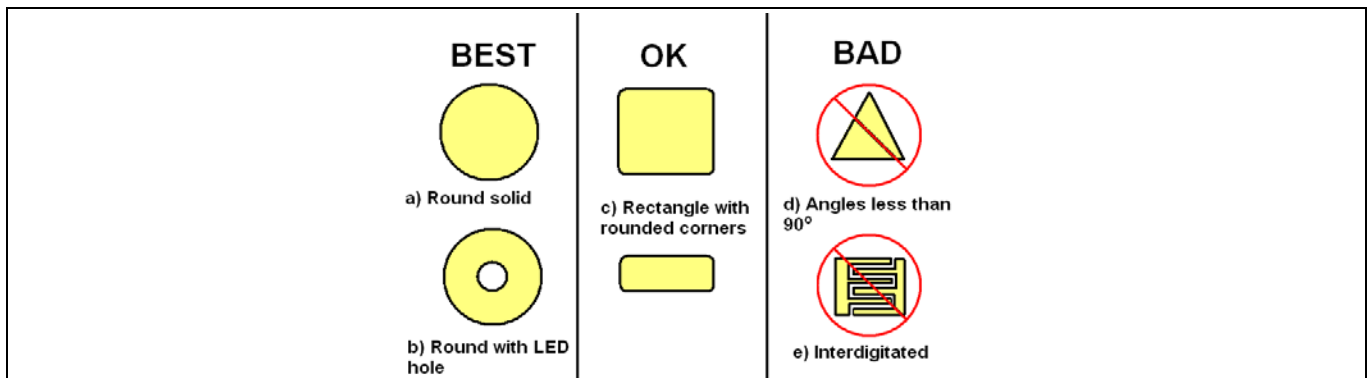


Figure 117 推奨ボタン形状

サイズ: ボタンの直径は 5mm～15mm の範囲でなければなりません。10mm がほとんどのアプリケーションに適します。より大きい直径は、より厚いオーバーレイに適します。

間隔: センサーパッドとグラウンドハッチ間の間隔は、オーバーレイの厚さに等しく、0.5mm～2mm の範囲でなければなりません。例えば、オーバーレイの厚さが 1mm なら、間隔は 1mm にしてください。しかし、オーバーレイの厚さが 3mm なら、間隔は 2mm にしてください。

Figure 118 に示すように、隣接するボタンの誤ったタッチ検出を防ぐために、ボタンに接触するとき、別のボタンとグラウンドハッチ間の間隔に近接しないように 2 つの隣接するボタン間の間隔を選択する必要があります。

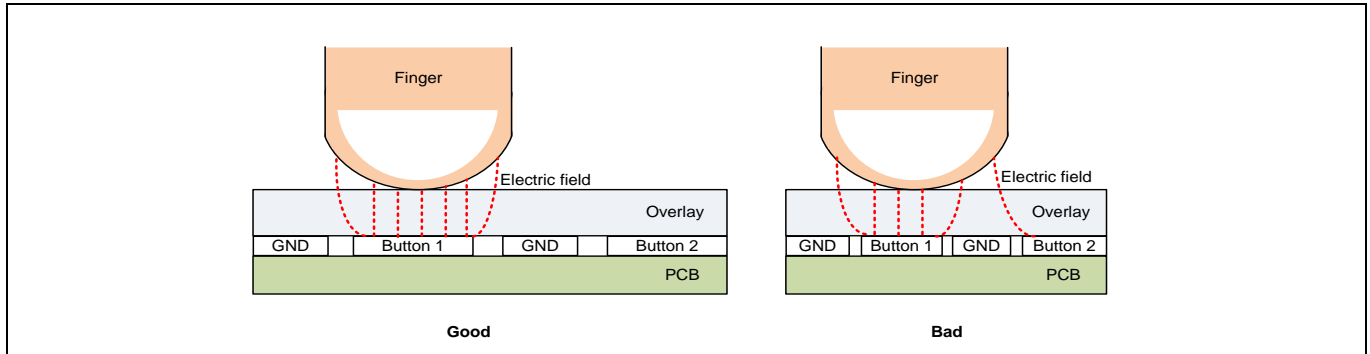


Figure 118 ボタン間隔

7.4.3.2 相互静電容量ボタンの設計

相互容量センシングは、2 つの電極間の容量結合の変化を測定します。センサーパターンは、指が Tx 電極と Rx 電極の間の電界を最大限に乱すように設計する必要があります。相互静電容量ボタンの設計に使用できる標準のボタンパターンがあり、それらのパラメーターはアプリケーションの要件に基づいて変更できます。フィッシュボーンパターンは、SNR の観点から優れたパフォーマンスを提供する相互容量パターンの 1 つです。

7.4.3.2.1 フィッシュボーンパターン

プロングまたはフィッシュボーンは、相互静電容量ボタンの標準形状です。Tx は、Rx をノイズから保護するために、ボタンの周りにボックスまたはリングを形成します。電界を形成するために、境界の内側に織り交ぜられた Tx および Rx プロングがあります。Figure 119 に、ハッチングされた地面を備えた

設計上の注意事項

上面図と底面図の 2 ピンフィッシュボーンセンサー構造の例を示します。Tx 電極の外壁と同一平面上のハッチグラウンドの間のギャップは、Tx および Rx 電極のエアギャップよりも大きくする必要があります。フィッシュボーン構造の基準面 (PCB 最下層) には、Figure 119 に示すようにボイド領域が必要です。

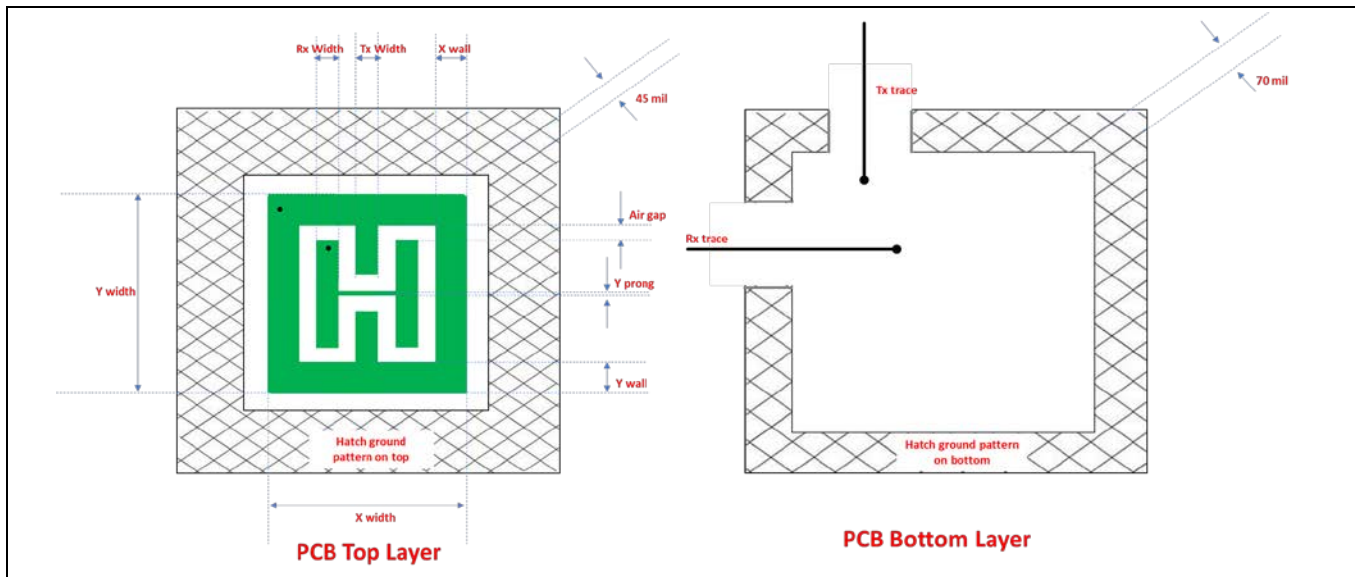


Figure 119 2つの Rx プロングを備えた CSX フィッシュボーンボタンパターン

Table 28 に、ガラスやアクリルなどの一般的に使用されるセンサーサイズとオーバーレイの推奨されるフィッシュボーンボタンの設計パラメーターを示します。センサーサイズのセクションで説明したように、推奨されるボタンサイズは、ボタンの X と Y の寸法を、予想されるユーザーの指のサイズとオーバーレイの厚さの合計に近づけることです。ただし、表には、ボード上の使用可能なスペースに制約がある場合、またはユーザーの操作を容易にするためにアプリケーション用に大きなボタンが必要な場合などに選択できる複数のボタンサイズが一覧されます。

また、特定のボタンサイズでは、オーバーレイの厚さが増すと、達成可能な SNR が低下することに注意してください。したがって、厚いオーバーレイ (約 1mm 以上のアクリルまたは 2mm のガラス) を使用する場合は、ボタンの SNR パフォーマンスがさらに制限されるため、ボードスペースによるボタンサイズの妥協を避けてください。このような厚いオーバーレイには、必ず大きなボタン (>= 予想される最大の指のサイズ) を使用してください。また、小さなボタンの場合は、SNR を向上させるために薄いオーバーレイを使用することを推奨します。

設計上の注意事項

Table 28 フィッシュボーンボタンの寸法 (すべての単位は mm)

ボタンサイズ (X 寸法, Y 寸法)(mm)	Rx プロング 数	Tx と Rx 間のエ アギャップ (mm)	Tx 幅 (mm)	Rx 幅 (mm)	X 壁幅 (mm)	Y 壁幅 (mm)	Y プロング (mm)
5, 5	3	0.35	0.48	0.48	0.24	0.24	0.2
10, 7	3	0.75	0.92	0.92	0.46	0.46	0.2
10, 5	3	0.5	1.17	1.17	0.58	0.58	0.2
10, 10	2	0.9	1.60	1.60	0.80	0.80	0.2
12, 12	2	1.3	1.70	1.70	0.85	0.85	0.2
13, 10	2	1.1	2.15	2.15	1.08	1.08	0.2
13, 13	2	1.5	1.75	1.75	0.88	0.88	0.2
15, 15	2	1.7	2.05	2.05	1.03	1.03	0.2
17, 17	2	2.3	1.95	1.95	0.98	0.98	0.2
20, 13	2	1.8	3.20	3.20	1.60	1.60	0.2
25, 13	2	2	4.25	4.25	2.13	2.13	0.2

この章の回路図とレイアウトのガイドラインに従うと、Table 28 の上記のボタン設計パラメーターにより良好な SNR パフォーマンスが保証されます。

Note: 設計でより高い外部ノイズが予想される場合、およびその他の複雑な場合は、ボタン設計の支援についてテクニカルサポートに連絡できます。外部ノイズの詳細については、CAPSENSE™ システムのノイズのセクションを参照してください。また、設計では、低い C_M が予想される場合は、それを軽減するために CSX ボタンの C_M が低いことを確認セクションに記載されるガイドラインに従ってください。

7.4.3.2.2 任意の形状と寸法のボタンデザイン

Figure 120 に、フィッシュボーンパターンでの Rx プロングのさまざまな方向を示します。ボタン A では、Rx プロングはボタンの側面に垂直で、寸法が大きくなります。ボタン B では、Rx プロングはボタンの側面に平行で、寸法が大きくなります。ボタン A のように Rx プロングを配置すると、ボタン B と比較してボタンパターンが最適化されます。したがって、Rx プロングは、ボタンの側面に対して垂直に、より大きな寸法にすることを常に推奨します。したがって、10x13mm のボタンが必要な場合は、Table 27 の 13x10 mm ボタンを使用し、90°回転させて、Figure 121 に示す 13x10mm の CAPSENSE™ システムのノイズのボタンパターンを取得します。

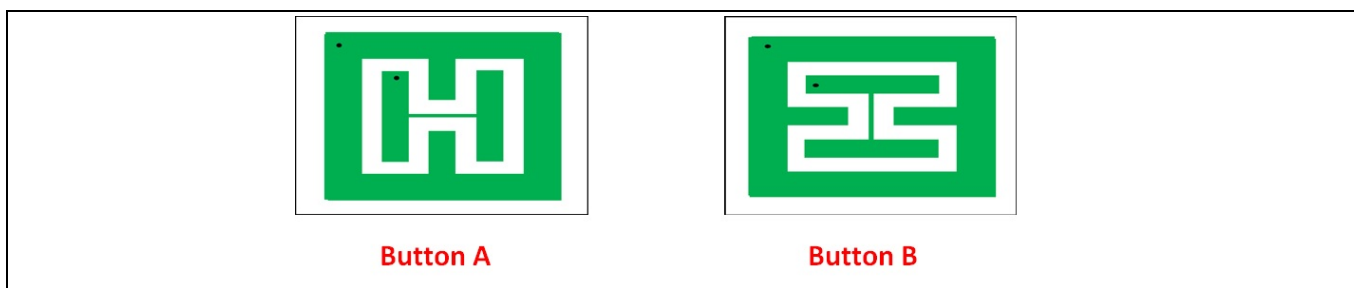


Figure 120 Rx プロングの向き

設計上の注意事項

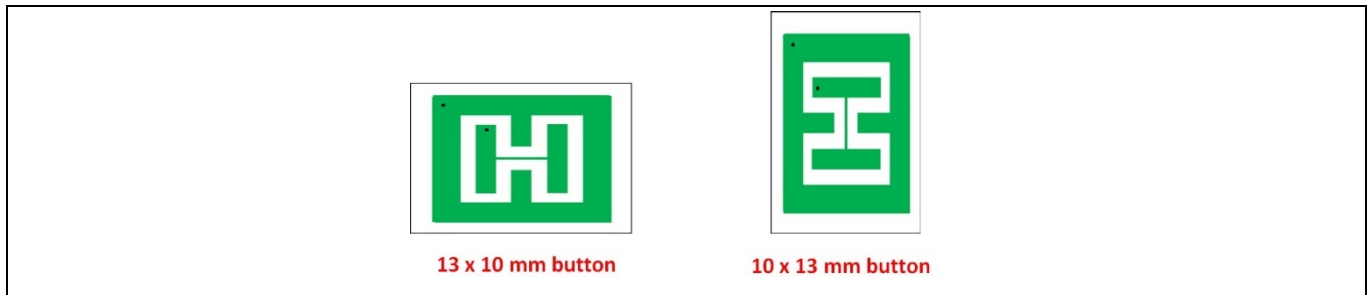


Figure 121 ボタンを 90 度回転させて、目的のボタンの寸法を取得

楕円形や円形など、推奨される長方形とは異なるボタンの形状が必要な設計があります。以下の手順では、標準のフィッシュボーンパターンから非標準の形状のボタンを作成する方法について説明します。

- 最初に、Table 27 からフィッシュボーンパターン (楕円形の場合は長方形、円形の場合は正方形ボタン) を選択して、目的のボタンの形状をカバーします。
- 次に、Figure 122 に示すように、ユーザーインターフェース上、またはフィッシュボーンパターン上に必要な寸法のオーバーレイ印刷ボタンの形状の上に配置します。

この方法を使用して設計された相互静電容量ボタンは、下の図に示すように、ボタンの形状の外側に過敏な領域または感度が低い領域があります。これは、相互静電容量ボタンのソフトウェアしきい値を適切に調整することで軽減できます。下の図に、正方形のフィッシュボーンパターンを使用して作成された円形のボタンの例を示します。

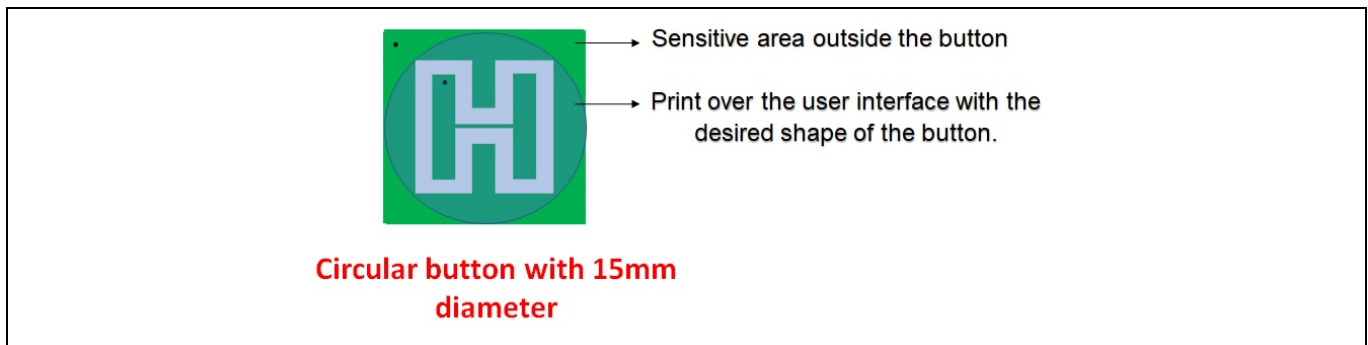


Figure 122 任意のパターンに基づく任意の形状のボタンデザイン

Table 27 にないパターンが必要な場合は、いくつかの手順に従ってボタンパラメーターを取得できます。例えば、19x19 のパターンが必要な場合は、17x17 のように Table 27 から必要なパターンに近いパターンを選択し、ボタン領域に関して Tx と Rx の間のエアギャップを測定します。以下に例を示します。

$$New_{gap} = TxRx_{gap} * \left(\frac{ButtonArea_{desired}}{ButtonArea_{reference}} \right)$$

ここで、ButtonArea = X x Y センサーの寸法です。

以下の仮定に基づいて、上記で取得した Newgap を考慮して、Tx 幅, Rx 幅, および Tx 壁を計算します。得られたボタン設計パラメーターの値を Table 28 に示します。ボタンの設計パラメーターの説明を理解するには、Figure 109 を参照してください。

設計上の注意事項

$$Tx_{width} = Rx_{width}$$

$$Tx_{wall} = \frac{Tx_{width}}{2}$$

Table 29 19 x 19 ボタン形状 17 x 17 ボタンのボタンパラメーター

ボタンサイズ (X 寸法, Y 寸法)(mm)	Rx プロング数	Tx と Rx 間のエア ギャップ (mm)	Tx 幅 (mm)	Rx 幅 (mm)	X 壁幅 (mm)	Y 壁幅 (mm)	Y プロング (mm)
17, 17	2	2.3	1.95	1.95	0.98	0.98	0.2
19, 19	2	2.9	1.85	1.85	0.93	0.93	0.2

7.4.3.2.3 フィッシュボーンパターンパラメーターに関する一般的な推奨事項

センサーサイズ

センサーサイズはボタンの XY 寸法であり、ボードスペースの可用性、予想されるユーザーの指のサイズ、オーバーレイの素材と厚さに基づいて選択されます。センサーサイズの選択は、PCB 内の必要なボタン間のギャップとスペースの可用性を考慮して、PCB 上の必要なボタンの数にも依存します。ただし、スペースが制約されない場合は、より高いボタンサイズを選択すると、良好な SNR が得られます。**Figure 123** に示すように、センサーのサイズを 1 ポイントを超えて大きくすると、SNR が飽和することに注意してください。これは、Tx/Rx 電極からの電界線の一部が指と相互作用しないためです。

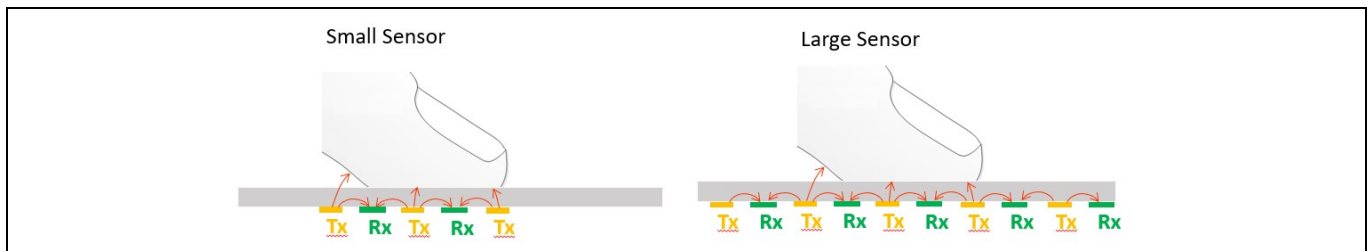


Figure 123 任意のパターンに基づく任意の形状のボタンデザイン

ボタンの SNR は、厚いオーバーレイを使用すると低下します。したがって、推奨される最小センサーサイズは、指のサイズとオーバーレイの厚さの合計であり、厚いオーバーレイでも良好な SNR を実現します。例えば、推奨される最小センサーサイズは、直径約 10 mm、オーバーレイの厚さ 3 mm の指のサイズを考慮すると、13x10 mm にできます。**任意の形状と寸法のボタンデザイン** で述べたように、Rx プロングは大きな寸法の側面に垂直である必要があります。

ボタンの間隔

ボタンの間隔は、2つのボタンの Tx 壁の間のギャップです。ボタンを相互に分離することでユーザーエラーを防ぎ、クロストークを減らします。ボタン間隔を 8 mm 以上に保つことを推奨します。これにより、シングルタッチとマルチタッチのパフォーマンスが向上します。

オーバーレイ

オーバーレイの厚さとオーバーレイの誘電率は、ボタンの SNR と、ESD などの外部ノイズに対する耐性に影響を与えます。詳細については、**オーバーレイの選択** セクションを参照してください。ボタンの SNR を高くするのに役立つオーバーレイの厚さをできるだけ小さくすることを推奨します。また、ESD ノイズに対する耐性を提供するのに十分な厚さである必要があります。場合によっては、オーバーレイの厚さが厚く、機械的な設計上の理由でそれを回避できない場合があります。このような場合、より良

設計上の注意事項

い SNR を得るには、推奨サイズよりも大きいサイズの相互静電容量ボタンを使用してください。オーバーレイの厚さに対するボタンの最小寸法の選択については、**センサーサイズ**のセクションを参照してください。より高い誘電率のオーバーレイ材料を使用すると、SNR も高くなります。したがって、厚いオーバーレイを使用する場合は、常に誘電率の高い材料を使用してください。また、ボタンが小さい場合は、SNR を向上させるために薄いオーバーレイを使用することを推奨します。

Tx 電極と Rx 電極間のエアギャップ

Tx 電極と Rx 電極の間のギャップは、Tx 電極と Rx 電極の間の相互容量に影響を与えます。ギャップを大きくすると、相互容量が減少します。これは、フィッシュボーンパターン設計で最も重要なパラメーターであり、750fF を超える相互容量を持つような Tx 電極と Rx 電極の間のギャップです。

Rx プロングの数

Rx プロングの数を増やすと、特定のボタンサイズの Tx 電極と Rx 電極の間のギャップが減少するため、Rx プロングの数は Tx 電極と Rx 電極間の相互容量に影響を与えます。より高い相互容量は、Tx 電極と Rx 電極の間のより高い電界線を意味します。したがって、ボタンに触れると、より高い信号が得られます。これは、指で触れると電界が最大限に乱されるためです。ただし、 C_M が高くなると、VDDA リップルノイズなどの外部ノイズの影響も大きくなります。したがって、より高い信号を取得するために Rx プロングの数を選択すること、優れたノイズ耐性を得ることはトレードオフがあります。フィッシュボーンパターン(つまり、1つの Tx プロングと2つの Rx プロングを備えたフィッシュボーンパターン)の場合、Rx プロングの最適な数は2です。次の図に、3つおよび1つの Rx プロングを備えた相互静電容量ボタンを示します。

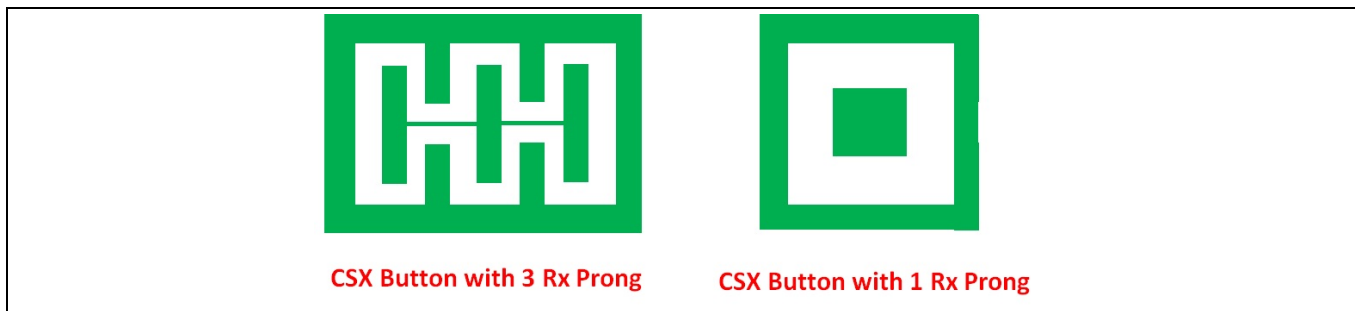


Figure 124 プロングの数が異なる相互静電容量ボタン

Tx 電極と Rx 電極の幅

Tx 電極と Rx 電極の幅は、Tx 電極と Rx 電極間の相互容量に影響を与えます。システム内の外部ノイズが少ない場合、Rx 幅/面積が Tx 幅/面積に等しいときに最良の信号応答が達成されます。次の図に、Tx 電極から Rx 電極までの幅が等しいまたは等しくない電界線を示します。したがって、下の図から、Tx と Rx の幅が等しいと、最終的に指で触れたときの C_M の変化が大きくなることが明らかです。

設計上の注意事項

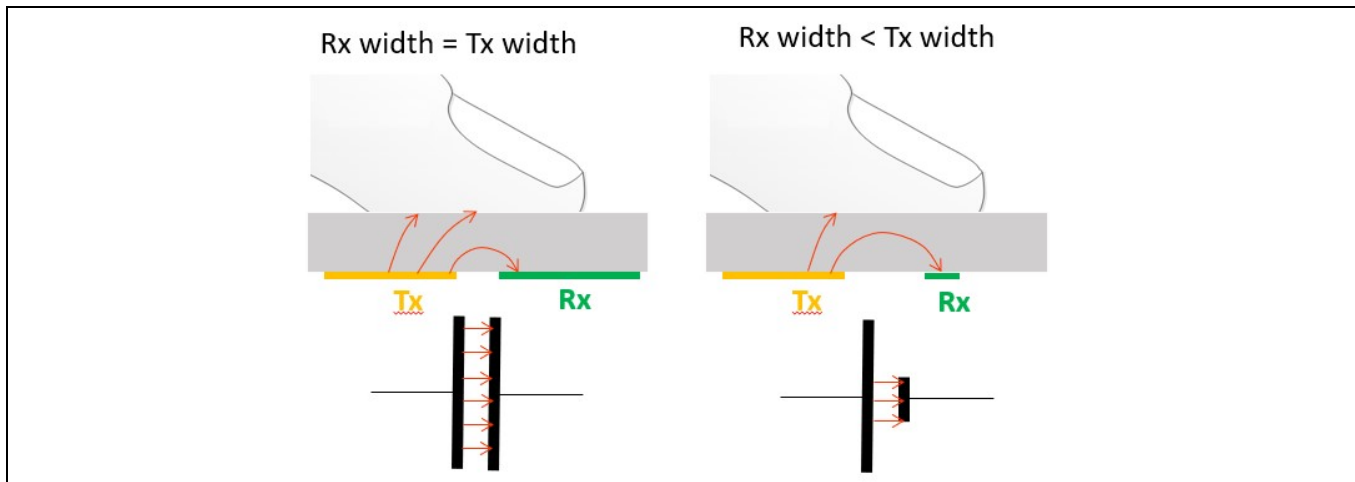


Figure 125 Tx 幅および Rx 幅の影響

場合によっては、**相互容量センシングの耐液性**のセクションで説明したように、相互容量ボタンに液体耐性を提供する必要があります。これを実現するには、CSX と CSD の両方のセンシング方法でハイブリッドセンシング技術を使用する必要があります。このような場合、CSD 技術でスキャンされた Tx または Rx 電極は、指で触れたときに良好な信号が得られるように、かなりの幅が必要です。

同一平面上のグラウンド

同一平面上のグラウンドが存在すると、システム内のノイズの影響が減少し、良好なグラウンドが提供されるため、信号の視差効果が減少します。センサーの周囲をハッチパターンでできるだけ多く取り、デバイスのアースに接続することを推奨します。また、この章のレイアウトと回路図のガイドラインに記載される推奨事項に従ってください。グラウンドプレーンは、接近する指への電界線の結合を減らし、指の接触によって引き起こされる相互静電容量の変化を減らします。センサーの真下に存在するノイズ源との強い結合が予想されない限り、センサーの下にグラウンドプレーンを配置しないことを推奨します。

Figure 119 に、PCB の最上層と最下層の同一平面上のグラウンドを示します。Tx 電極の外壁と同一平面上のハッチグラウンドの間のギャップは、Tx および Rx 電極のエアギャップよりも大きくする必要があります。

Tx 壁 (X 壁と Y 壁の幅)

Tx 壁は、ノイズから Rx 電極へのシールドとして機能します。広い Tx 壁は、クロストークの影響と同一平面上のグラウンドの影響も軽減します。Tx 壁幅を Tx 電極幅の半分にほぼ等しく保つことを推奨します。下の図に、より広い Tx 壁の効果を示します。これは、同一平面上のグラウンドの影響を減らすことにより、Tx 電極から指に到達する電界線の数を増やします。**相互容量センシングの耐液性**のセクションで説明したように、Tx 電極を CSD センサーとしてスキャンする場合は、Tx 壁の幅をわずかに増やせます。Table 29 に、Tx ウォールを増やした 10x10 パターンの例を示します。

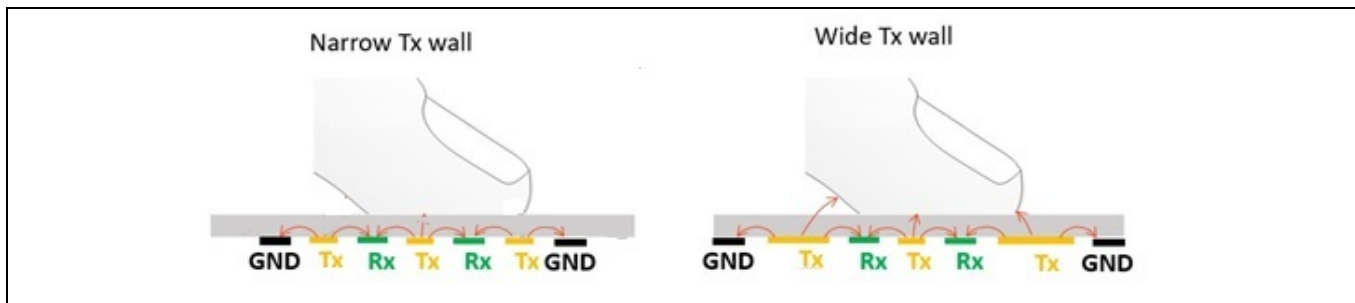


Figure 126 Tx 壁幅の影響

設計上の注意事項

Table 30 Tx 壁を増やした 10x10 ボタンの寸法 (すべての単位は mm)

ボタンサイズ (X 寸法, Y 寸法)(mm)	Rx プロング数	Tx と Rx 間のエア ギャップ (mm)	Tx 幅 (mm)	Rx 幅 (mm)	X 壁幅 (mm)	Y 壁幅 (mm)	Y プロング (mm)
10, 10	2	0.8	1.2	1.2	1.5	1.6	0.2

7.4.4 スライダーの設計

Figure 127 にリニア スライダーのための推奨スライダー パターンを示し、Table 31 にそれぞれのリニア スライダー寸法の推奨値を示します。推奨レイアウトのガイドラインの詳細は、次の節で説明します。

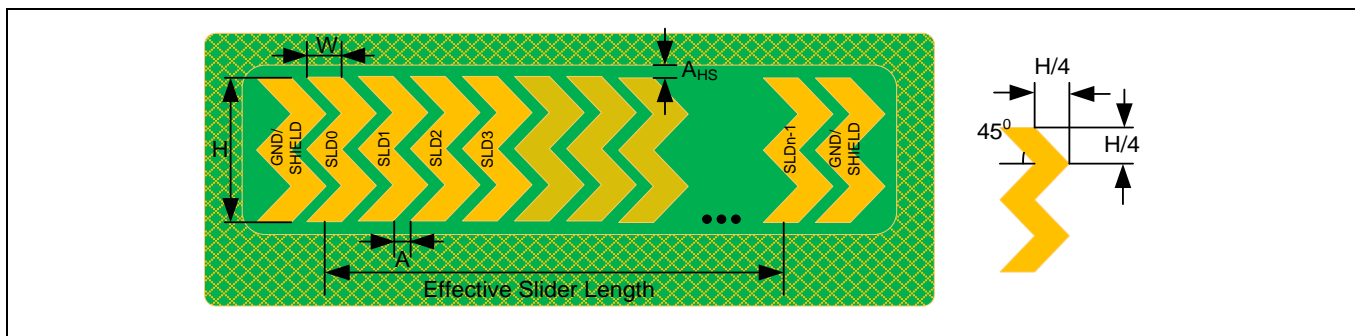


Figure 127 標準リニアスライダーパターン

Table 31 リニアスライダーの寸法

パラメーター	アクリル オーバーレイの厚さ	最小	最大	推奨値
セグメントの幅 (W)	1mm	2mm	-	8mm ¹¹
	3mm	4mm	-	
	4mm	6mm	-	
セグメントの高さ (H)	-	7mm ¹²	15mm	12mm
セグメント間のエアギャップ (A)	-	0.5mm	2mm	0.5mm
ハッチとスライダー間のエアギャップ (A _{HS})	-	0.5mm	2mm	オーバーレイの厚さに等しい

¹¹ 推奨されるスライダーセグメントの幅は人間の指の平均直径 9mm に基づいて計算します。詳細は [スライダーセグメントの形状、幅およびエアギャップ](#) を参照してください。

¹² 推奨されるスライダーセグメントの最小の高さ 7mm は、人間の指の最小直径 7mm に基づき推奨されるものです。オーバーレイの厚さと CAPSENSE™ チューニングにより、指があらゆるセグメントの真ん中に置かれるときに (信号対ノイズ比 ≥ 5:1) を達成できるという条件で、スライダーの高さが 7mm 未満にできます。

設計上の注意事項

7.4.4.1 スライダー セグメントの形状、幅およびエアギャップ

スライダー上の実際の指の位置に対する報告された指の位置 (すなわち、重心位置) のリニア応答を実現するためには、指がスライダー セグメントの真ん中ではなく、セグメント SLD0 の真ん中とセグメント SLDn-1 の真ん中の間の任意の位置に置かれたときに、ちょうど 2 つのセンサーが有効な信号を報告するようにスライダーを設計することが必要です¹³。指が任意のスライダー セグメントの真ん中に置かれた場合、隣接するセンサーは、(差分カウント=ノイズしきい値)と報告されなければなりません。そのため、Figure 127 に示すように、二重の山形状を使用することを推奨します。この形状は、Figure 128 および Figure 129 に示すように、理想的な応答に近い重心応答を達成するために有用です。同様の理由で、スライダーセグメント幅とエアギャップ (Figure 127 でマークされる、それぞれ寸法「W」と「A」) は、式 38 で述べた関係に従うべきです。

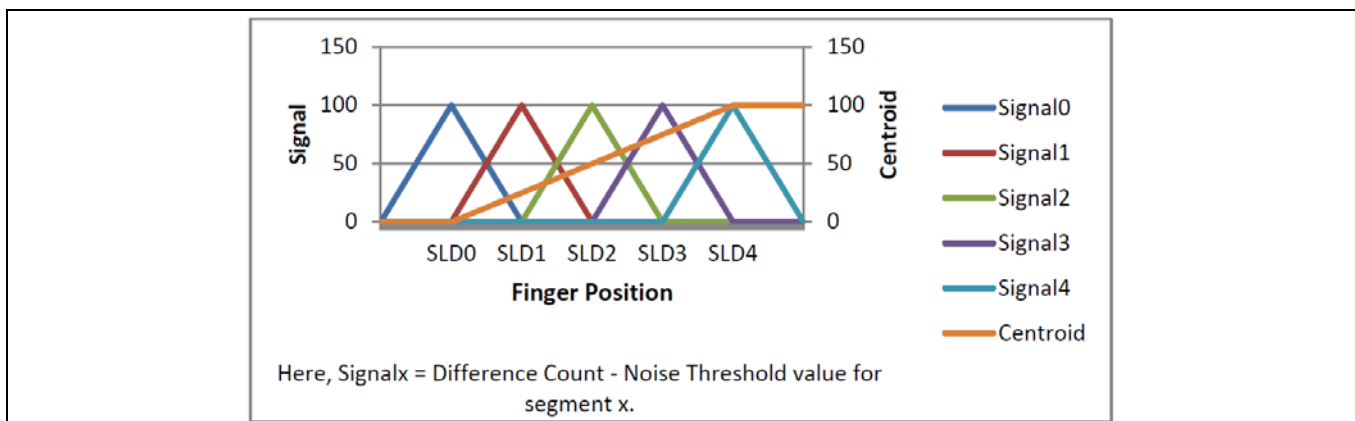


Figure 128 理想的なスライダーセグメント信号と重心応答

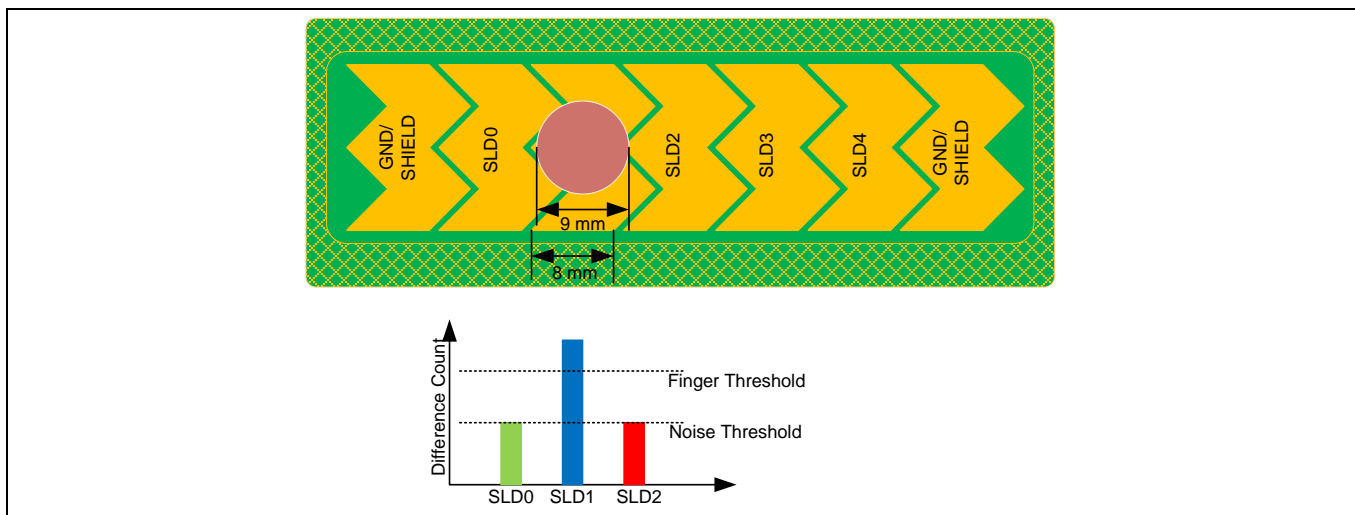


Figure 129 理想的なスライダー信号

¹³ここで、有効な信号は特定のスライダー セグメントの差分カウントがノイズしきい値以上であることを意味します。

設計上の注意事項

式 73. セグメント幅、エアギャップと指の直径の関係

$$W + 2A = \text{指の直径}$$

一般的には、人間の指の平均直径は約 9mm です。この指の平均直径と式 73 に基づいて計算すると、推奨のスライダーセグメント幅とエアギャップはそれぞれ 8mm と 0.5mm になります。

(スライダーセグメント幅+2*air-gap) が指の直径よりも小さい場合、式 73 に示した関係に応じて、重心応答が非線形になります。この場合、これは指をスライダーに置くと、Figure 130 に示すようにスライダーで静電容量が増加し、3 つ以上のスライダーセグメントで与えられた位置で有効な信号が発生するためです。このように、式 74 によって計算された重心位置は Figure 131 のように非線形になります。

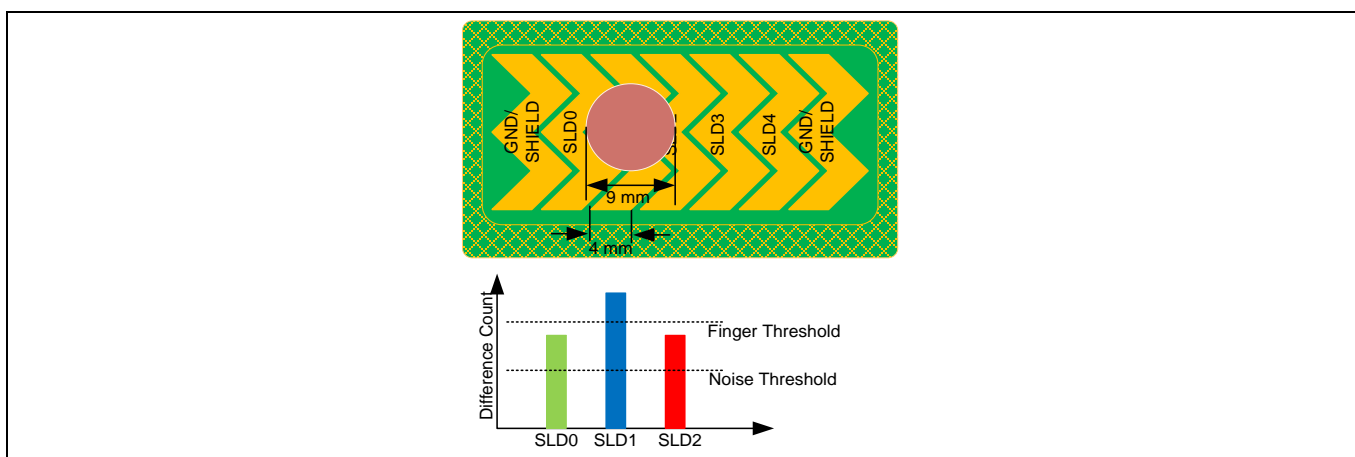


Figure 130 スライダーのセグメント幅が推奨値よりも狭いときに、指は 3 つ以上のセグメント上で有効な信号を発生

式 74. PSoC™ Creator で CAPSENSE™ コンポーネントにより使用される重心アルゴリズム

$$\text{重心位置} = \left(\frac{S_{x+1} - S_{x-1}}{S_{x+1} + S_x + S_{x-1}} + x \right) * \frac{\text{分解能}}{(n - 1)}$$

ここで、

分解能 = CAPSENSE™ コンポーネント カスタマイザーで設定された API 分解能

n = CAPSENSE™ コンポーネント カスタマイザー内のセンサー要素の数

x = 最大信号を与える要素のインデックス

S_i = スライダーセグメントの差分カウント (ノイズしきい値を除く)

設計上の注意事項

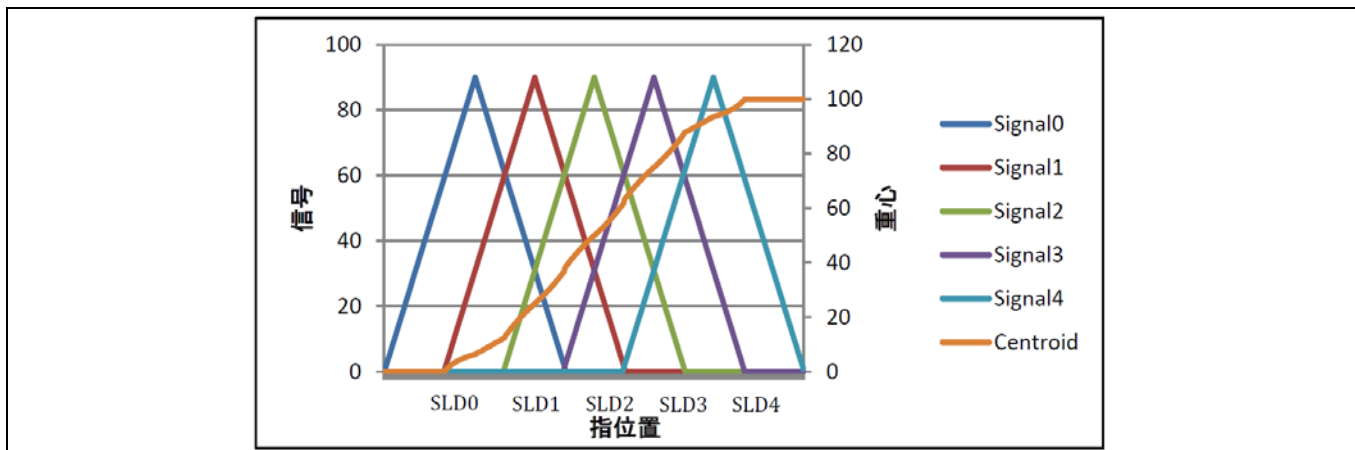


Figure 131 スライダー セグメント幅が推奨値より低いときの非線形重心応答

Figure 131 に示すように、*slider-segment-width* の値が (*finger diameter*-2**air-gap*) よりも小さい場合、重心応答が非線形になりますが、実際の指の位置に対する報告される重心応答の線形性が重要な役割を果たさない最終アプリケーションに使用される場合があります。ただし、有効なスライダー長上の任意の位置では、少なくとも1つのスライダーセグメントが(信号対ノイズ比 $\geq 5:1$) (すなわち、信号が指しきい値パラメーター以上)を提供するように、スライダーセグメント幅の最小値はオーバーレイの厚さに基づいて維持されなければなりません。スライダーセグメント幅が小さすぎると、指が静電容量を十分に発生できない場合があるため、どのスライダーセグメントも (5:1) の SNR を達成できません。結果として、Figure 132 に示すように、重心値が PSoC™ Creator では 0xFFFF¹⁴、ModusToolbox™では 0x0000¹⁵と報告されます。

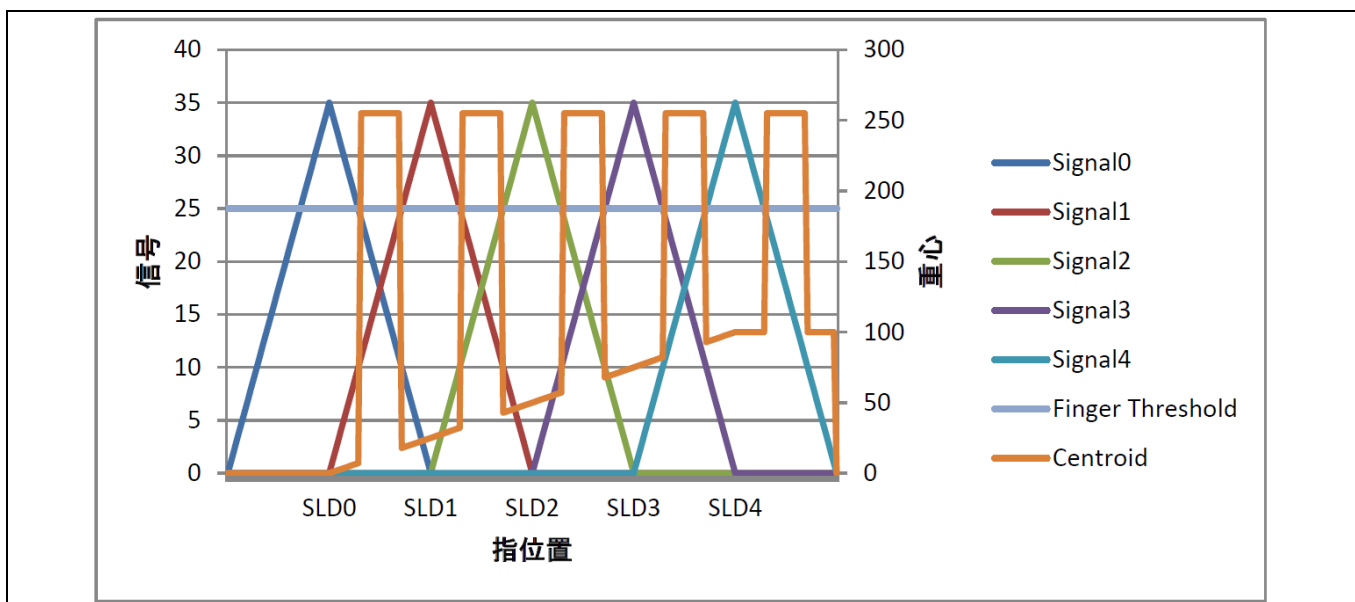


Figure 132 スライダー セグメント幅が小さすぎたときに報告された不正な重心

¹⁴ スライダー上で指が検出されなかった場合、またはスライダーセグメントのいずれにおいても差分カウント値が指しきい値パラメーターを超えない場合、PSoC™ Creator 内の CAPSENSE™コンポーネントは重心値を 0xFFFF と報告します。

¹⁵ スライダー上で指が検出されなかった場合、またはスライダーセグメントのいずれにおいても差分カウント値が指しきい値パラメーターを超えない場合、ModusToolbox™内の CAPSENSE™ミドルウェアは重心値を 0x0000 と報告します。

設計上の注意事項

アクリル オーバーレイの場合、特定のオーバーレイの厚さの値に対応するスライダー セグメント幅の最小値を、Table 31 に示します。Table 31 に記載していないアクリル オーバーレイの厚さの値については、Figure 133 を使用して最小スライダー セグメント幅を見積もれます。オーバーレイが非常に薄い場合、オーバーレイがない場合、raw カウントの飽和または指の静電容量の増加により、非線形の重心応答が発生する可能性があります。スライダーに触れる前に重心位置が検出される場合があります。これらの条件では、CAPSENSE™ 重心アルゴリズムは、式 74 を使用してスライダー上の指の位置を正しく推定できません。Table 30 に記載されているように、CSD センサーのオーバーレイの厚さを設定することを推奨します。

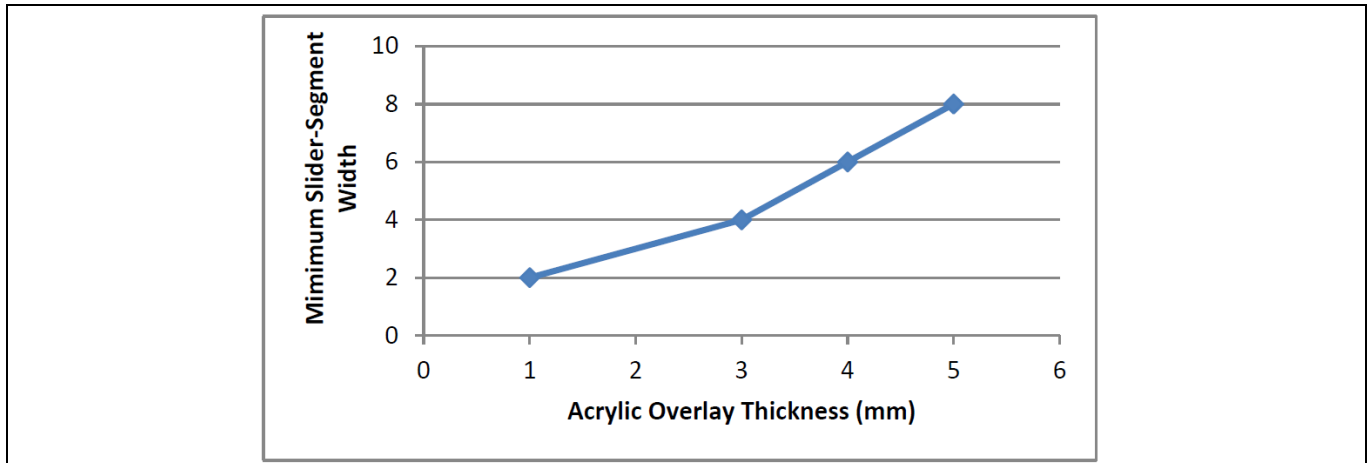


Figure 133 アクリル オーバーレイ用のオーバーレイの厚さに対応する最小スライダー セグメント幅

($slider-segment-width + 2 * air-gap$) が指の直径よりも大きい場合には式 73 に示した要件に応じて、重心応答はフラットスポットの形になります (すなわち、Figure 134 に示すように指がいずれかのセグメントの中央付近で移動しても、報告される重心位置が不変)。これは、Figure 135 に示すように、指がスライダー セグメントの真ん中に置かれた場合、指が隣接セグメントに向かって少し移動しても、その部分のみに有効な信号を追加されるためです。

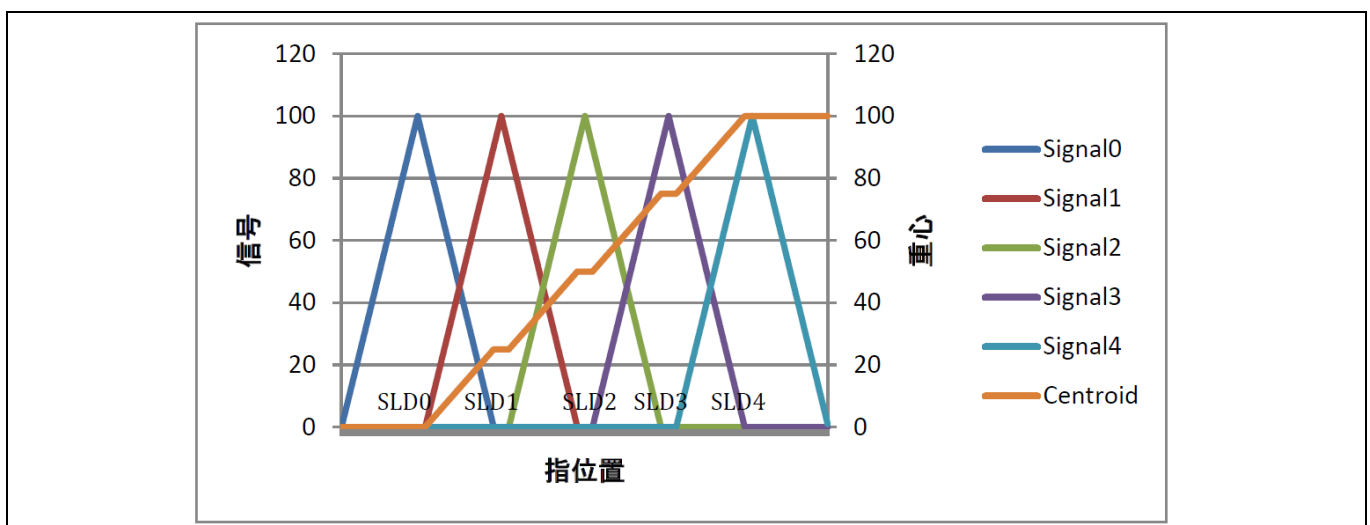


Figure 134 スライダー セグメント幅が推奨値より大きい場合のフラットスポット (非応答重心)

設計上の注意事項

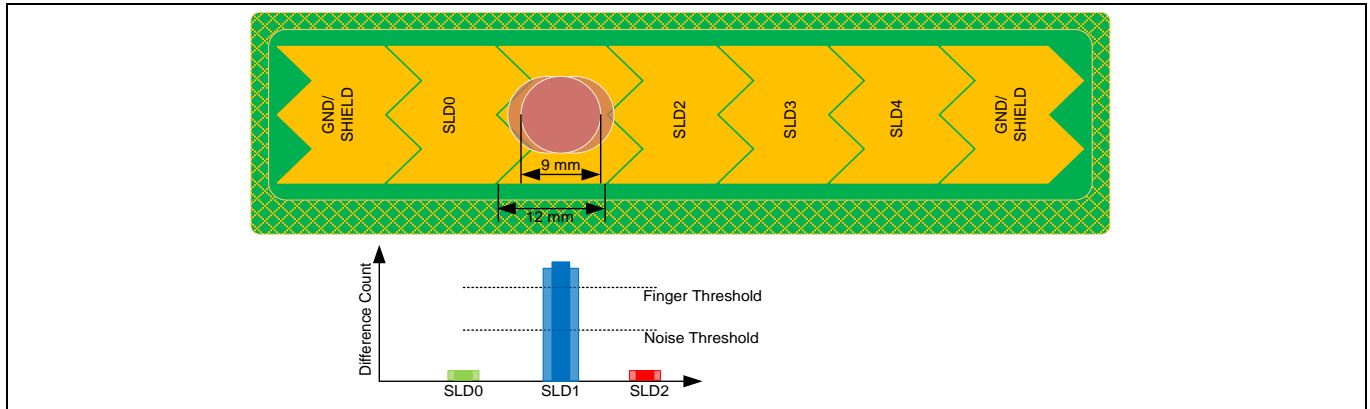


Figure 135 スライダー セグメント幅が推奨値より大きい場合のスライダー セグメント上の信号

(slider-segment-width+2*エアギャップ) が指の直径よりも大きい場合、指がスライダー セグメントの中央に置かれても、隣接するセンサーは、ノイズしきい値に等しい差分カウント値を報告するように、すべてのスライダー セグメントの感度を向上させ調整できるように注意してください (Figure 128 を参照してください)。しかしそうすると、ホバー効果が発生します (すなわち、指がただスライダーの上に停止しており、スライダーにタッチしなくても、スライダーは重心位置を報告する場合があります)。

7.4.4.2 スライダーの端におけるダミー セグメント

CAPSENSE™ 設計では、1つのセグメントがスキャンされると、隣接するセグメントは、CAPSENSE™ CSD コンポーネントの「非アクティブ センサー 接続」パラメーターで指定されるオプションによりグラウンドまたは被駆動シールド信号のいずれかに接続されます。リニア重心応答では、スライダーのすべてのセグメントが同じ感度を持つ必要があります。すなわち、指をスライダーのセグメントに置いたときの raw カウント (信号) の増加は、すべてのセグメントで同じでなければなりません。すべてのスライダー セグメントから均一の信号レベルを維持するためには、スライダーの両端の2つのセグメントをグラウンドまたは被駆動シールド信号のいずれかに物理的に接続することが推奨されます。グラウンドまたは被駆動シールド信号への接続は、「非アクティブ センサー 接続」パラメーターで指定される値に依存します。そのため、アプリケーションが n 個セグメントのスライダーを必要とする場合、Figure 127 に示すように、(n+2) 個の物理的なセグメントの作成を推奨します。

空間の制限のため、スライダーの両端にそれぞれに1つのセグメントを配置できない場合は、Figure 136 に示すように、これらのセグメントを最上層のハッチに実装できます。また、使用可能な空間の全領域がまだ制約されている場合も、これらのセグメントの幅がセグメント SLD0~SLDn-1 の幅よりも小さくするか、またはこれらのダミーセグメントも取り除けます。

スライダーの両端の2つのセグメントが最上層のハッチに接続している場合、最上層のハッチを「Inactive sensor connection」パラメーターで指定される信号に接続する必要があります。スライダーに耐液性が必要な場合、スライダー周囲のハッチ、最後の2つのセグメントおよび非アクティブなスライダーセグメントは被駆動シールド信号に接続する必要があります。詳細は [自己容量センサーに対する液滴と液体流の影響](#) を参照してください。

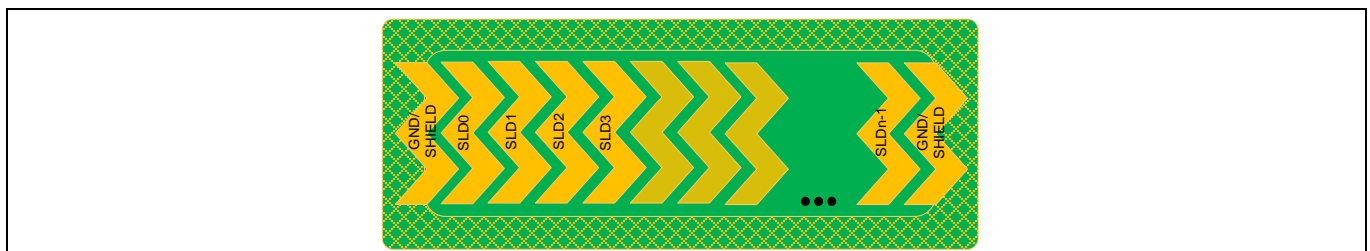


Figure 136 最初と最後のセグメントが最上層のハッチに接続されるリニア スライダーのパターン

設計上の注意事項

7.4.4.3 スライダー寸法の決定

特定のデザインのスライダー寸法は以下の考慮事項に基づいて選択できます。

- アプリケーション要件に基づいて、スライダーの必要な長さ (L) を決定してください。これは **Figure 127** に示す「有効なスライダー長」と同じです。
- 基板上の使用可能な空間に応じてセグメントの高さを決定してください。基板スペースが対応できる場合、最大許容セグメントの高さ (15mm) を使用します。そうでない場合、より低い高さを使用しますが、使用する高さは **Table 31** に指定された最小値より大きいことを確保してください。
- スライダーセグメント幅とスライダーセグメント間のエアギャップを **Table 31** に示す推奨値にしなればなりません。指の平均直径 9mm の場合、推奨スライダーセグメント幅とエアギャップはそれぞれ 8mm と 0.5mm です。
- 特定のスライダー長さ (L) は、以下の式を用いて必要なセグメントの数を計算してください。

式 75: スライダーに必要なセグメント数

$$\text{セグメント数} = \frac{\text{スライダー長さ}}{\text{スライダーセグメント幅} + \text{エアギャップ}} + 1$$

スライダーを実装するためには少なくとも 2 つのスライダーセグメントが必要であることに注意してください。

特定のアプリケーションでは、CAPSENSE™ ピンに利用できる数が計算されたセグメントの数よりも少しだけ少ない場合は、利用できるピン数で目標のスライダーの長さを達成するために、セグメント幅を増やせます。例えば、10.2cm のスライダーには 13 個のセグメントが必要です。しかし、10 ピンだけが利用可能である場合、セグメント幅を 10.6cm に大きくできます。これにより、**Figure 134** に示すように応答が非線形になったり、ホバー効果が生じます。しかし、このレイアウトは最終アプリケーションで高い線形性を必要としない場合には使用できます。

Figure 127 に示すように PCB の長さは必要なスライダーの長さより大きいことに注意してください。PCB の長さは、**式 76** に示すようにスライダーの長さに関係します。

式 76. PCB 最小の長さ と スライダーの長さの関係

$$\text{PCB 長さ} = \text{スライダー長さ} + 3 * \text{スライダーセグメント幅} + 2 * \text{エアギャップ}$$

PCB の利用可能な空間の大きさが上記の式に示す要件よりも小さい場合は、ダミーセグメントを取り除けます。

この場合、PCB の必要な最小長は **式 77** に示すようになります。

式 77. 最小 PCB 長さ と スライダー長 の関係

$$\text{PCB 長さ} = \text{スライダー長さ} + \text{スライダーセグメント幅}$$

7.4.4.4 スライダーセグメント配線の経路

単一のスライダーは多くのセグメントがあり、各セグメントはデバイスの CAPSENSE™ 入力ピンに独立して接続されます。各セグメントは個別にスキャンされ、重心位置を計算するために重心アルゴリズムは最後にすべてのセグメントの信号の値に適用されます。SmartSense アルゴリズムはスライダーに特別なチューニング方法を実装しており、セグメントの C_p 差異によって起こりえる重心の非直線性を回避します。しかし、スライダーが動作するために、以下のレイアウト条件を満たす必要があります。

- セグメントの C_p は常に **コンポーネントデータシート** に記載されている C_p の範囲内にある必要があります。

設計上の注意事項

- スライダセグメントの C_p は可能な限り近くする必要があります。ただし、実際のシナリオでは、各セグメントのトレースルーティングが異なるため、各スライダセグメントの C_p が異なる場合があります。セグメント寄生容量の最大許容変動は、85% IDAC キャリブレーションレベルのスライダセグメントの最大 C_p 44% です。 C_p の変動がこの制限を超えると、スライダセグメント間の感度に変化し、スライダの応答が非線形になる可能性があります。

線形応答の優れたスライダ設計を満たすように、以下のレイアウト設計基準で実装します。

- できるだけ均一になるように、すべてのセグメント形状を設計
- セグメントをデバイスに接続する配線の長さや幅は、可能ならすべてのセグメントでは同じであることを確保
- センサー/配線とグランド面/ハッチ間に同一のエアギャップを維持

7.4.4.5 LED 付きスライダ デザイン

いくつかのアプリケーションでは、LED 駆動で指の位置を表示する必要があります。**Figure 137** に示すように、スライダセグメントの上に LED を配置するか、または LED バックライト用にスライダセグメントの真ん中に穴が開けられます。LED を配置するために穴を開けると、スライダセグメントの有効面積が減ります。**SNR** は 5:1 以上に達成するために、スライダセグメントの幅は LED の穴の大きさよりも大きくなければなりません。特定のオーバーレイ厚さ用に **SNR** は 5:1 以上に達成するために、必要な最小スライダ幅については **Table 31** を参照してください。**クロストークに対するソリューション** で説明するガイドラインに従って LED を配線してください。

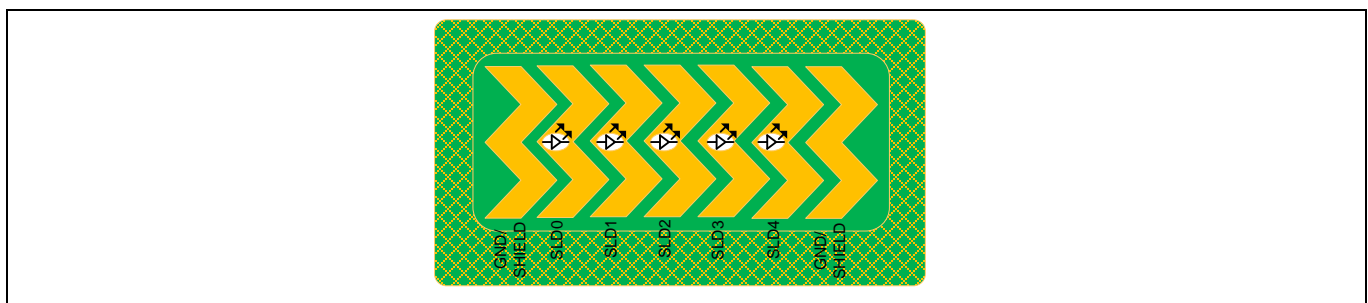


Figure 137 LED バックライト付きスライダ設計

7.4.5 センサーとデバイスの配置

PCB 設計でセンサーおよびデバイスを配置する際は、これらのガイドラインに従ってください。

- デバイスピンからセンサーパッドへの配線の長さは最小にしてください。
- 直列抵抗をデバイスピンの 10mm 以内に配置して、RF 干渉を低減し、ESD 保護を行ってください。詳細は **CAPSENSE™ ピンの直列抵抗** を参照してください。
- デバイスおよび他の部品を PCB の最下層に搭載してください。
- コネクタが C_p とノイズの回り込み (noise pickup) を増加するため、センサーとデバイスピンの間にはコネクタを配置しないでください。
- ボタン間の距離 (端から端まで) は 8mm より大きくする必要があります。キー間の距離が 8mm 未満の場合、キー間のクロストークが発生します。また、使いやすさの観点から、ユーザーが 2 つのキーに同時に触れるリスクが高まります。キー間の距離は 8mm より大きくなければなりません
- タッチラインから金属までの間隔は 5mm より大きくする必要があります。これには、金属製のシャーシ、装飾用のクロームトリム、ネジなどが含まれます。

設計上の注意事項

- CAPSENSE™ コンポーネントとその信号を、変圧器などのノイズの多いサブシステムから分離または物理的な距離をとってください。一般に、CAPSENSE™ システムは外部ノイズに敏感です。

7.4.6 配線の長さと同幅

センサーの寄生容量を最少にするために、短くて狭い PCB 配線を使用します。推奨される最大の配線の長さは、標準的な PCB の場合は、12 インチ (300 mm) で、フレックス回路の場合は 2 インチ (50 mm) です。推奨される最大の配線幅は 7 mil (0.18 mm) です。CAPSENSE™ の配線をハッチングドグランドあるいはハッチシールドで囲み、配線とハッチの間隔を 10 mil~20 mil (0.25 mm~0.51 mm) けてください。

7.4.7 配線の経路

センサー配線を PCB の最下層で配線する必要があります。そうすると、指は配線に作用しません。センサーに接続していない配線は、センサーパッドの真下で配線しないでください。

スイッチング信号または通信ラインの 0.25mm よりも近くで静電容量センシング配線を配置しないでください。センシング配線と他の信号間の距離を増やすと、ノイズ耐性が増えます。通信ラインおよびセンサーピンが交差する必要がある場合、交差部が **Figure 138** で描かれているように直角であることを確認してください。

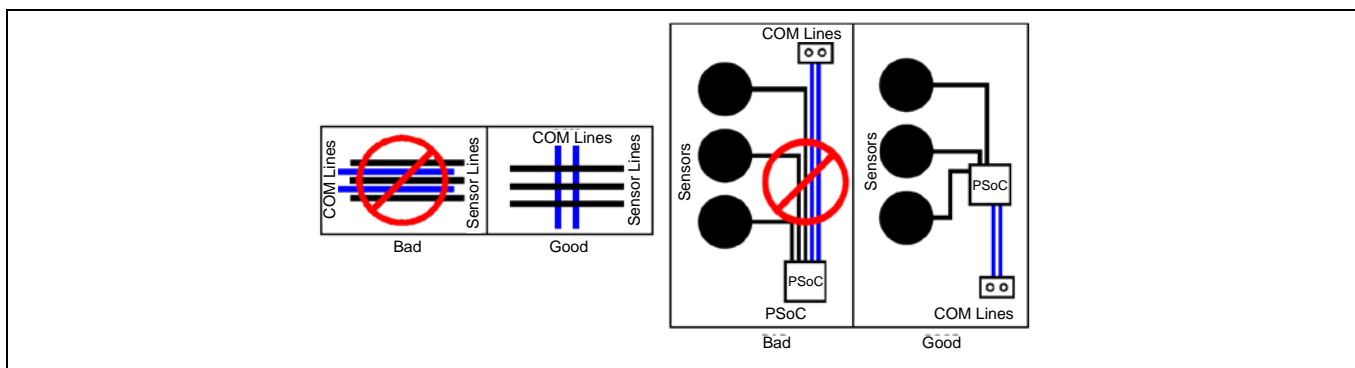


Figure 138 センサーと通信ラインの配線経路

スペース制約により、センサー配線は I²C 通信ラインまたは Bluetooth® LE アンテナ配線などの高速の配線と並行に取り付けられる場合、**Figure 139** に示すようにセンサー配線と高速の配線との間にグランド配線を配置することが推奨されます。このガイドラインも、高精度アナログ配線 (温度センサーから PSoC™ デバイスまでの配線など) の CAPSENSE™ センサー配線により発生するクロストークに適用されます。グランド配線の厚さは 7mils であることが可能であり、センサー配線からグランド配線までの間隔は最低 10mil にして CAPSENSE™ センサーの C_p を削減します。

設計上の注意事項

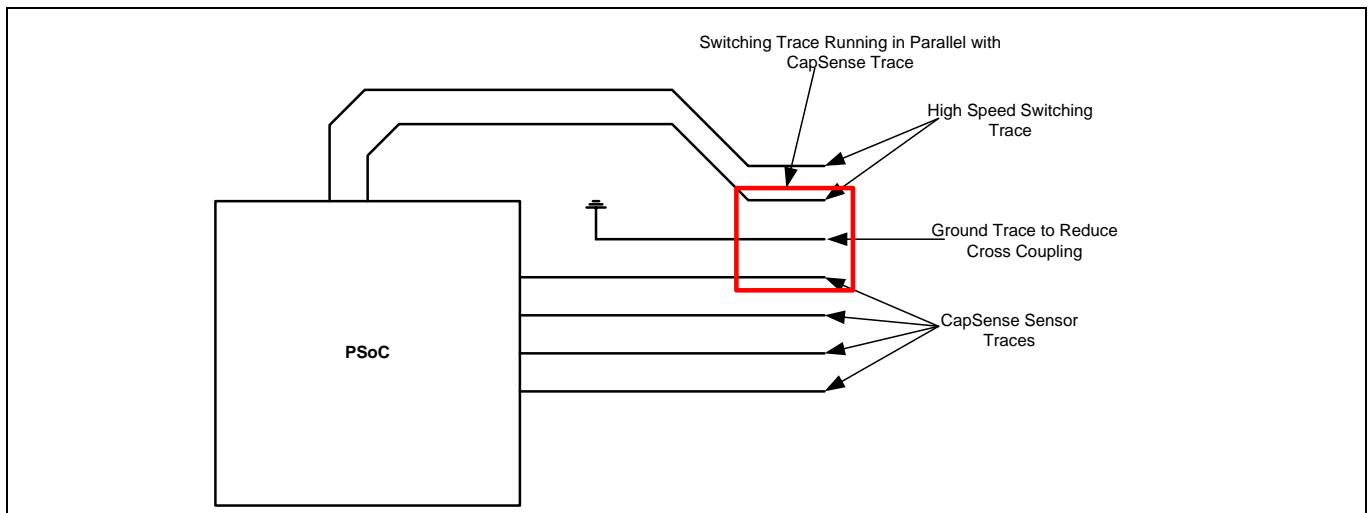


Figure 139 高速のスイッチング配線と CAPSENSE™配線間のクロストークを削減

グランド配線がスイッチング配線と CAPSENSE™配線との間に配置できない場合、3W ルールに従って、配線間のクロストークを削減します。3W ルールでは、Figure 140 に示すように、「隣接する配線のクロストークを低減するために、エッジ間に最低2つの配線幅の間隔を維持する必要がある」と述べられます。

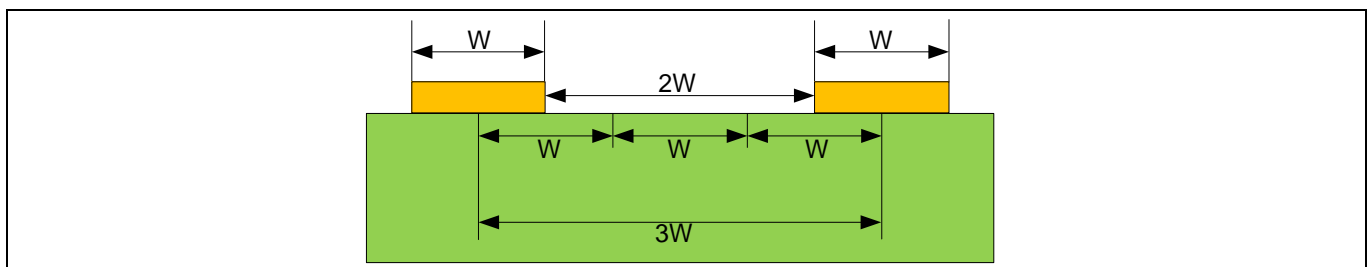


Figure 140 クロストークを最低限するために 3W ルールに従う配線間隔

- Tx ラインと Rx ラインを互いに平行に走らせないでください。トレースルーティングは可能な限り分離する必要があります。
- レイアウトの制約により Tx と Rx が短距離で平行に走る必要がある場合、Tx と Rx の間のスペースをキー内の Tx と Rx の間の距離よりも大きくする (Tx-Rx キー間隔の2倍が望ましい) か、それらの間にグランドを追加する必要があります。
- タッチキーのノイズを防ぐために、Rx の周囲にはできるだけ多くのクリアランスを確保してください。電源配線と LED ライン (高速スイッチング、電源) の間隔については、このガイドラインに従うことが重要です。グランドもこの規則に従う必要がありますが、それほど重要ではありません。アースはノイズ保護を提供しますが、キーの感度を低下させます。
- 特定のセンサーセットの場合、Rx ラインの数は Tx ライン以下である必要があります。Rx ラインはノイズの影響を受けやすいのに対し、Tx ラインは比較的影響を受けにくいです

7.4.8 クロストークに対するソリューション

パネル用の一般的なバックライト技術は、センサーパッドの中央にある穴を通して見られるように、センサーパッドの下に LED を搭載することです。LED がオン/オフに切り替わる場合、LED 配線上の電圧の変化により容量センサー入力にクロストークが発生し、センサーデータにノイズが混入する可能性があります

設計上の注意事項

あります。このようなクロストークを回避するために、**配線の経路**セクションで説明したとおりに CAPSENSE™ を LED 配線から分離します。

また、**Figure 141** に示すように、フィルターを使用して、LED 駆動電圧の急速な変化を避けることによりクロストークも低減できます。必要な LED 応答速度に応じてフィルターを設計します。

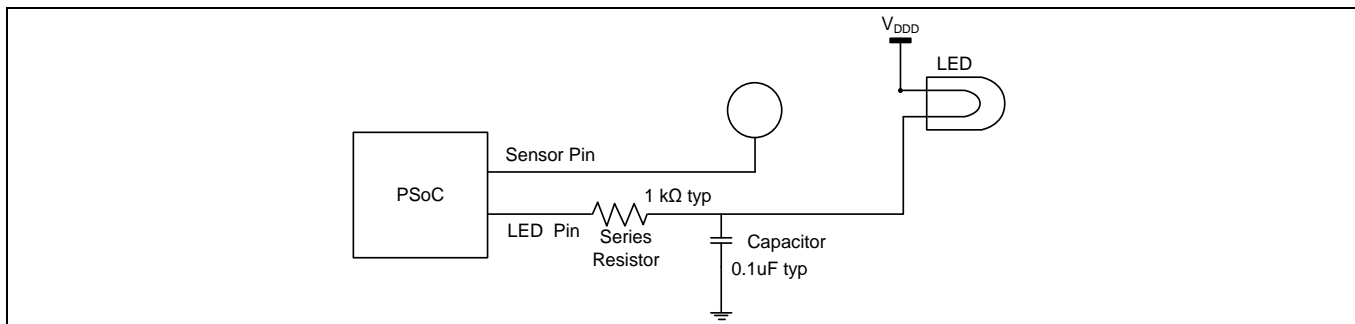


Figure 141 クロストーク削減

ガード配線は、相互容量ボタンの Tx/Rx ラインの近くまたは上/下を走るグランド配線です。レイアウトでグランドハッチが許可されていない場合は、ガード配線を使用してセンサー配線をノイズから保護できます。グランドハッチと同様に、ガード配線は寄生容量を追加し、ボタンの感度を低下させます。ガード配線は通常、ケースバイケースで必要です。過去にガード配線が使用された典型的な状況は次のとおりです。

- クロストークの削減
- 高速ライン (I2C, SPI, UART) のノイズや LED 配線の切り替えからの保護
- HMI または LCD の周囲の境界線

7.4.9 ビア

最少数のビアを使って CAPSENSE™ 信号を配線し、寄生容量を最小にしてください。**Figure 142** に示すように、配線長を短縮するためにビアをセンサーパッドの端に配置してください。

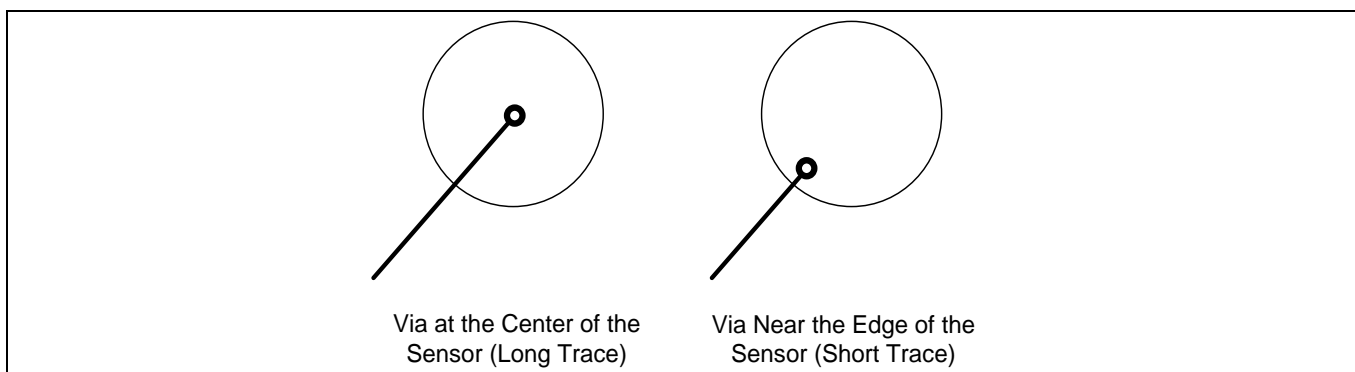


Figure 142 センサーパッド上のビアの配置

7.4.10 グランド面

グランド面を設計するとき、以下のガイドラインに従ってください。

設計上の注意事項

- センサーを囲むグランド面はハッチパターンにする必要があります。PCBの最上層と最下層の両方にグランド面を使用する場合、最上層にハッチング 25% (7mil ライン、45mil 間隔) を、最下層にハッチング 17% (7mil ライン、70mil 間隔) を使用する必要があります。グランド面の代わりに被駆動シールドを使用する場合、最上層と最下層の両方に同じハッチを使用します。
- CAPSENSE™ と関係がない基板の他の部品に対しては、ベタグランドをできるだけ多く使用する必要があります。
- PCB 製造コストに応じて、異なる階層上のグランド面はできる限り一緒にステッチする必要があります。ステッチがより多いと、小さいグランドインダクタンスになり、チップグランドが電源グランドに近づきます。無線が使用されるようなグランド経由の高いシンク電流がある場合は、これは特に重要です。
- CAPSENSE™ に使用されるグランド面はすべて中心点にスター接続する必要があります、この中心点は電源グランドへの唯一のリターン経路である必要があります。特に以下は重要です。
 - すべてのセンサー用のハッチンググランドは中心点で終了する必要があります。
 - C_{MOD} 、 C_{INTX} 用のグランド面は中心点で終了する必要があります。
 - C_{SH_TANK} 用のグランド面は中心点で終了する必要があります。

Figure 143 にスター接続を示します。各ファミリの中心点は Table 32 に示します。

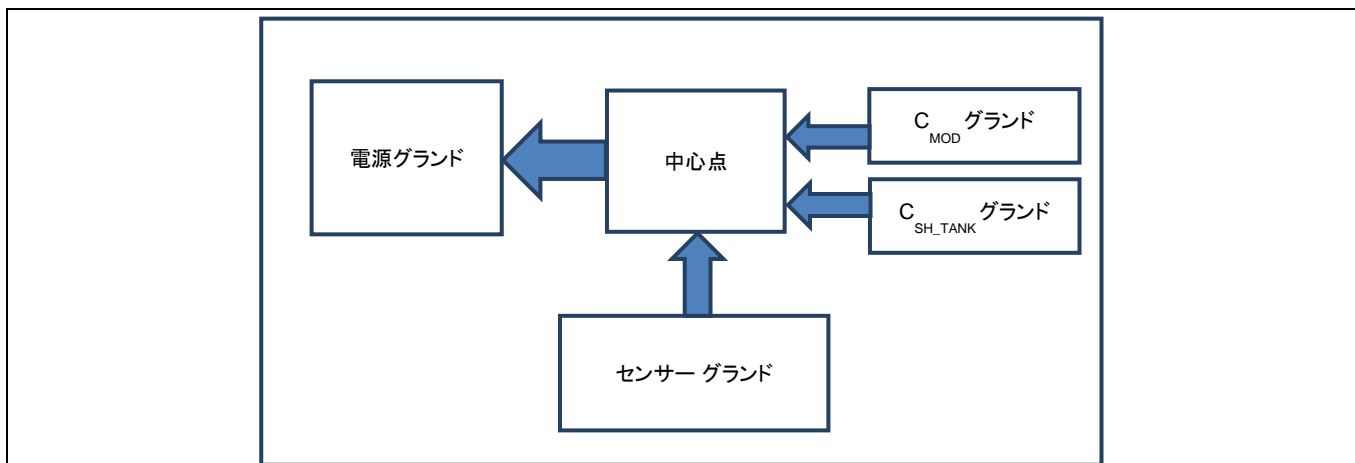


Figure 143 グランド面のスター接続

Table 32 スター接続の中心点

ファミリ	中心点
PSoC™ 4000	VSS ピン
PSoC™ 4100/4100M	VSS ピン
PSoC™ 4200/4200M/4200L/PSoC™ 4-S/PSoC™ 4100PS	VSS ピン
PSoC™ 4100-BL	E-pad
PSoC™ 4200-BL	E-pad

- CAPSENSE™ 用のすべてのグランド面は中心点から 0.2nH 以下のインダクタンスである必要があります。これを達成するためには、 C_{MOD} 、 C_{INTX} 、および C_{SH_TANK} コンデンサパッドをチップの近くに配置し、それらのグランド面を十分な厚さにします。

設計上の注意事項

7.4.10.1 E-pad のないパッケージの使用

E-pad を使用しない場合、VSS ピンは中心点であり、電源グランドへの唯一のリターン経路である必要があります。

E-pad のないチップを使用する場合、基板の最上層と最下層の高レベルのレイアウト図は [Figure 144](#) および [Figure 145](#) に示します。

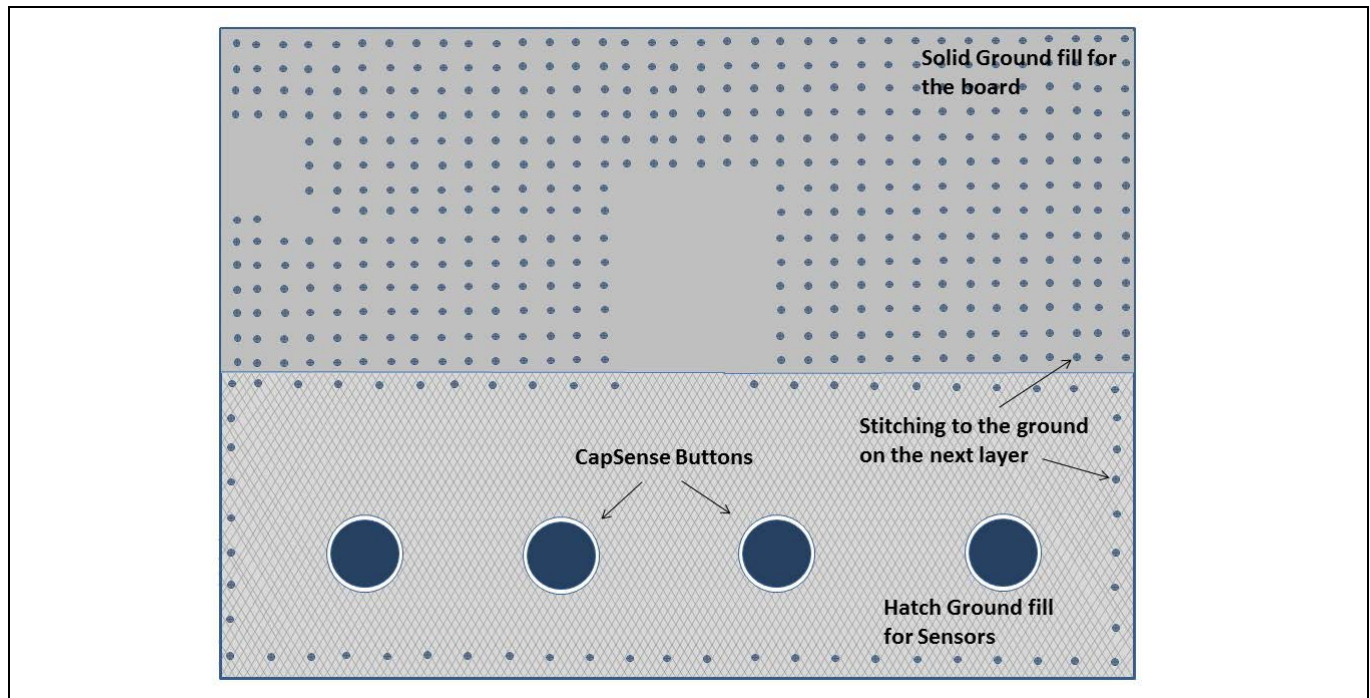


Figure 144 E-pad がないチップを使用する場合の PCB 最上層のレイアウト

設計上の注意事項

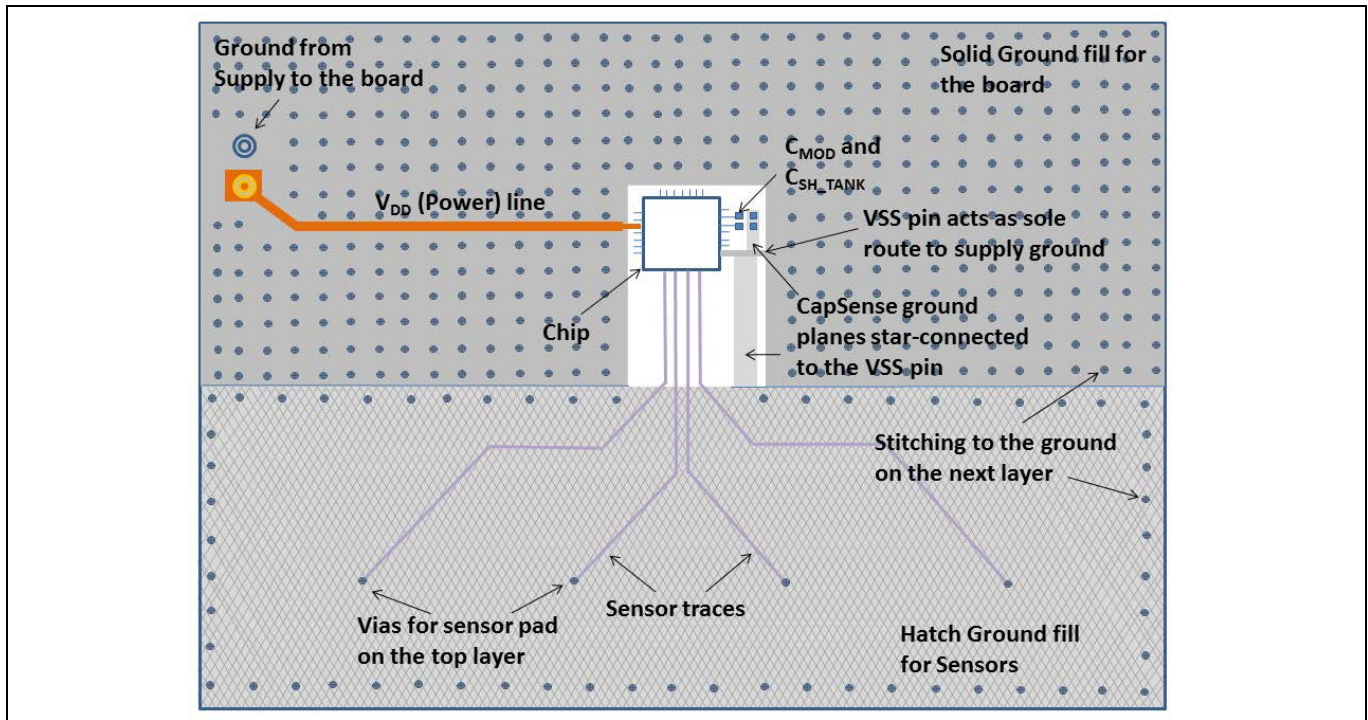


Figure 145 E-pad がないチップを使用する場合の PCB 最下層のレイアウト

7.4.10.2 E-pad のあるパッケージの使用

E-pad のあるパッケージを使用する場合、以下のガイドラインに従ってください。

- E-pad は中心点であり、電源グランドへの唯一のリターン経路である必要があります。
- E-pad は追加のグランドのために次の階層に接続するために下部にビアを持つ必要があります。通常、フィルしないビアはコスト目的のため設計で使用されますが、銀-エポキシでフィルされるビアを使用すればグランド経路で最低インダクタンスという結果になり、最高性能のために使用することが推奨されます。

7.4.10.3 PSoC™ 4 Bluetooth® LE デバイスの使用

QFN パッケージでの PSoC™ 4 Bluetooth® LE デバイス (E-pad あり) の場合

- グランド面の上述の一般的なガイドラインに従います。
- **E-pad のあるパッケージの使用**での E-pad 使用ガイドラインに従います。
- VSSA ピンはチップ下部の E-pad に接続する必要があります。
- E-pad 下部のビアは、10mil サイズの 5x5 ビアにすることが推奨されます。

PSoC™ 4 Bluetooth® LE チップを使用する場合の、基板の最上層と最下層の高レベルのレイアウト図を **Figure 146** および **Figure 147** に示します。

設計上の注意事項

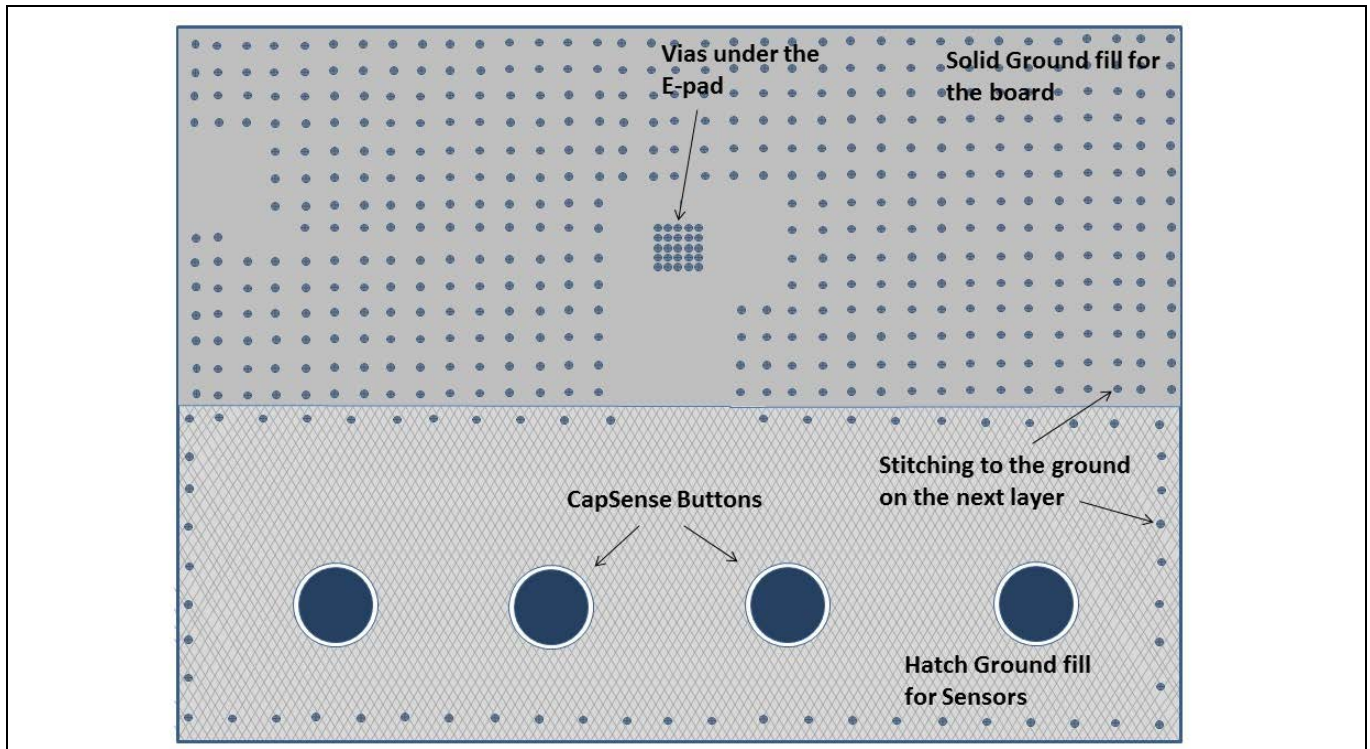


Figure 146 PSoC™ 4 Bluetooth® LE を使用する場合は PCB 最上層レイアウト (E-pad あり)

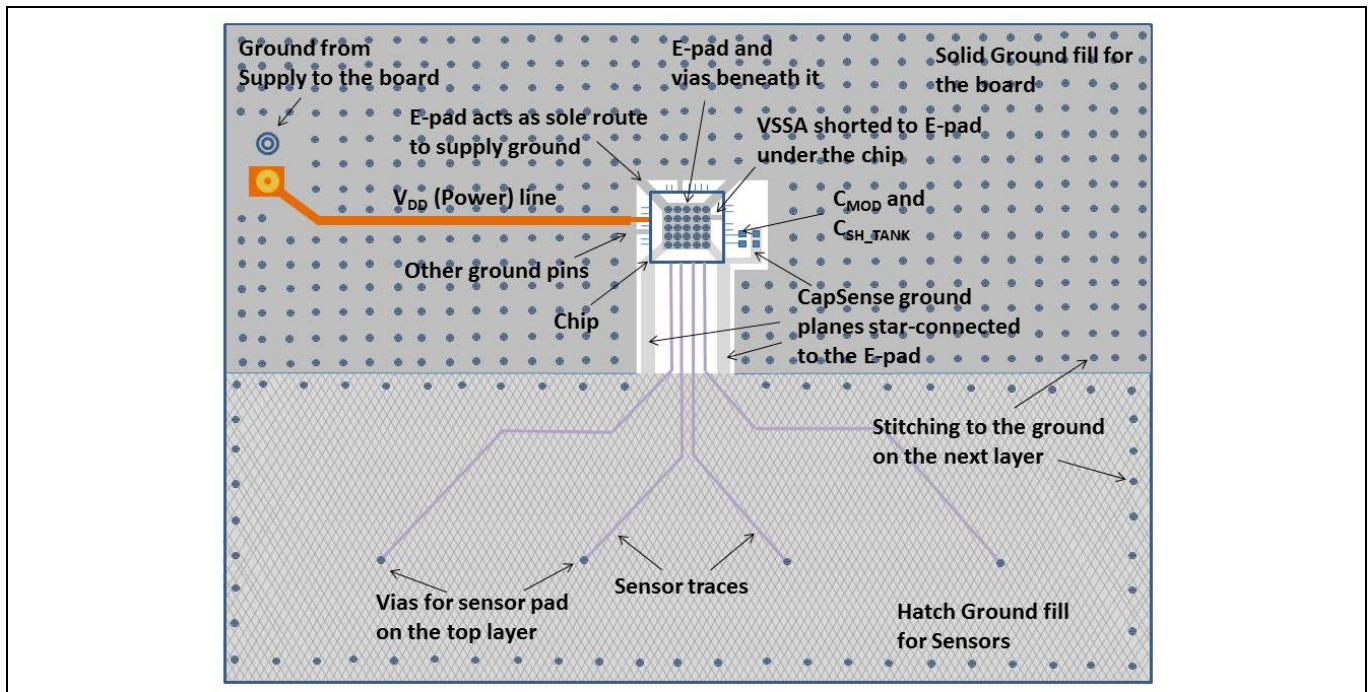


Figure 147 PSoC™ 4 Bluetooth® LE を使用する場合は PCB 最下層レイアウト (E-pad あり)

7.4.11 電源レイアウトの推奨事項

CAPSENSE™は高感度アナログシステムです。そのため、不適切な PCB レイアウトにより、近接センサーと厚いオーバーレイのボタン(1mmより厚い)などの高感度センサー構成でノイズを発生させます。

設計上の注意事項

高感度の CAPSENSE™ 設計で低ノイズを達成するためには、Table 33 に示すように、PCB レイアウトは電源ライン上にデカップリング コンデンサを持つ必要があります。

Table 33 電源ラインにおけるデカップリング コンデンサ

電源ライン	デカップリング コンデンサ	対応するグランド端子	適用可能なデバイス ファミリ
VDD	0.1µF および 1µF	VSS	PSoC™ 4000
VDDIO	0.1µF および 1µF	VSS	PSoC™ 4000, PSoC™ 6 MCU
VDDD	0.1µF および 1µF	VSS	PSoC™ 4100, PSoC™ 4200, PSoC™ 6 MCU
	0.1µF および 1µF	VSSD	PSoC™ 4100-BL, PSoC™ 4200-BL, PSoC™ 4200L, PSoC™ 4S シリーズ, PSoC™ 4100S Plus, PSoC™ 4100S Max
VDDA ¹⁶	0.1µF および 1µF (バッテリーで動作する場合)	VSSA	PSoC™ 4100, PSoC™ 4200, PSoC™ 4100-BL, PSoC™ 4200-BL, PSoC™ 4200L, PSoC™ 4S シリーズ, PSoC™ 4100S Plus, PSoC™ 4100PS, PSoC™ 6 MCU
	0.1µF および 10µF (主電源で動作する場合)	VSSA	PSoC™ 4S シリーズ, PSoC™ 4100S Plus, PSoC™ 4100PS
VDDR	0.1µF および 1µF	VSSD	PSoC™ 4100-BL, PSoC™ 4200-BL, Bluetooth® LE 接続搭載の PSoC™ 6 MCU
VCCD	デバイス データシートを参照してください。	VSS (PSoC™ 4000) または VSSD (他のすべてのデバイス)	すべてのデバイス ファミリ

グランド インピーダンスと電源配線の長さを最低にするために、デカップリング コンデンサと C_{MOD} コンデンサをできるだけチップの近くに配置する必要があります。

バイパス コンデンサの詳細は [デバイス データシート](#) の電源の節を参照してください。

7.4.12 耐液性のレイアウト ガイドライン

耐液性 セクションで説明したように、シールド電極とガード センサーを実装することにより、耐液性がある CAPSENSE™ システムを実現できます。デバイスに複数の CSD ブロックがある場合、各 CSD ブロックには専用のシールド電極が必要です。ここでは、シールド電極とガード センサーを実装する方法について説明します。

¹⁶ PSoC™ 4S シリーズ、PSoC™ 4100S Plus および PSoC™ 4100PS ファミリの V_{DDA} ピンは電源供給源に応じてバルク コンデンサの異なる値を必要とします。デバイスはバッテリーで動作する場合、0.1µF および 1µF のコンデンサを並列に使用することが推奨されますが、デバイスが主電源で動作する場合、0.1µF および 10µF のコンデンサを並列に使用することが推奨されます。これは CAPSENSE™ ブロックで使用されるリファレンス生成器 (REFGEN) の電源電圧変動除去比を改善するためのものです。

設計上の注意事項

7.4.12.1 シールド電極のレイアウトガイドライン

シールド電極の領域は、液滴の大きさおよびシールド電極を実装するための基板上の利用可能な領域に依存します。シールド電極はセンサーパッドと配線を取り囲み、これらから 1cm 以内に広げる必要があります。シールド電極を 1cm 超えて広げても、システム性能への影響はほとんどありません。

また、大きいシールド電極は放射エミッションが増加する可能性があります。基板が非常に広い場合は、**Figure 148** に示すように、1cm のシールド電極の外側の領域は、何も無いようにする必要があります。ボードの設計では、液滴とグランド間の結合容量を減らすことに重点を置く必要があります。したがって、耐液性を向上させるために、PCB の最上層と最下層にハッチフィルやグランドに接続されたトレースがあってははいけません。

グランドに接続されたハッチ (ベタ) や配線がある場合、液滴がタッチ面に当たると、センサーに誤ったトリガーを発生させることがあります。センサーとグランド間にシールド電極が配置されていても、シールド電極の効果は完全にマスクされ、センサーは誤ってトリガーする可能性があります。

アプリケーションによっては、PCB のシールド電極の実装に十分な領域がないことがあります。このような場合、シールド電極を 1cm 以内で広げられます。センサー実装後の基板上的残りの領域を最小限のシールド電極のエリアにできます。

一部のアプリケーションでは、シールド電極の静電容量が非常に高くなります。以下の手法で減らせます。

- 単一のシールド電極の代わりに複数のシールド電極を使用する: C_p が高い単一のハッチパターンがある場合は、ハッチパターンを複数のハッチパターンに分割し、シールド信号で駆動してシールド C_p を下げてください。これにより、センサーに高い範囲のセンスクロック周波数を使用できるようになり、CAPSENSE™ システムの感度が向上します。複雑なレイアウト設計では、このアプローチによりトレースルーティングが簡単になります。
- 複数のシールドピンを同じ電極に接続する: レイアウトでシールド電極を分割することが不可能な場合は、複数のシールドピンを同じ電極に接続してください。これにより、センサーピンのすべての直列抵抗が並列になり、シールド電極の有効時定数が減少します。そして、センサーに高い範囲のセンスクロック周波数を使用できるようになり、CAPSENSE™ システムの感度が向上します。

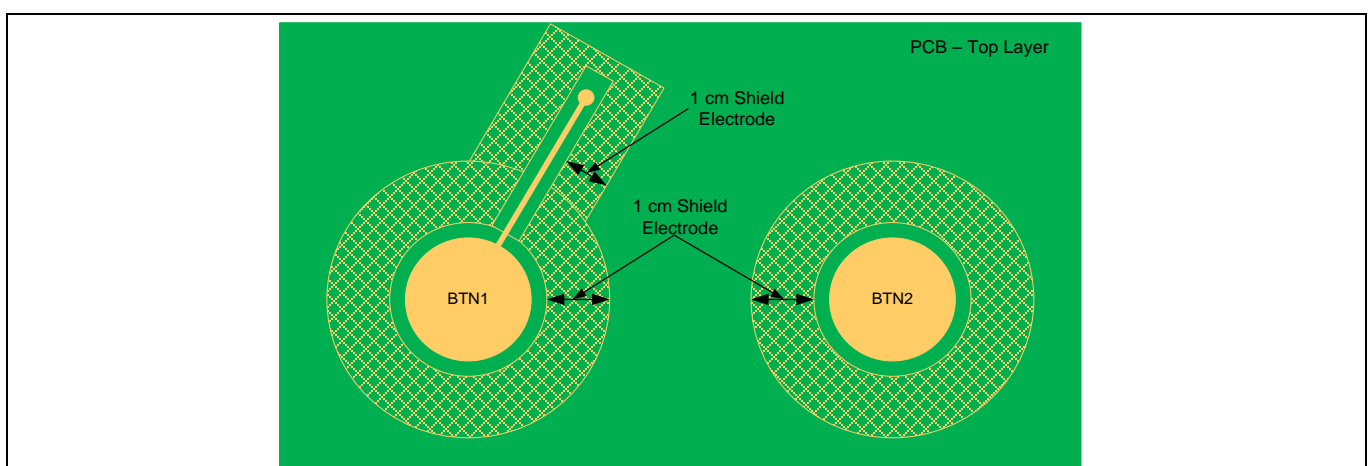


Figure 148 センサー配線が最上層と最下層で配線されるときにシールド電極の配置

2 層と 4 層の PCB にシールド電極を実装するためには以下のガイドラインに従ってください。

設計上の注意事項

2 層 PCB の場合

- 最上層: 7mil 配線と 45mil グリッドのハッチ (25%フィル)。ハッチは被駆動シールド信号に接続する必要があります。
- 最下層: 7mil 配線と 70mil グリッドのハッチ (17%フィル)。ハッチは被駆動シールド信号に接続する必要があります。

4 層 (またはそれ以上) PCB の場合

- 最上層: 7mil 配線と 45mil グリッドのハッチ (25%フィル)。ハッチは被駆動シールド信号に接続する必要があります。
- 第 2 層: 7mil 配線と 70mil グリッドのハッチ (17%フィル)。ハッチは被駆動シールド信号に接続する必要があります。
- 第 3 層: VDD 面
- 最下層: 7mil 配線と 70mil グリッドのハッチ (17%フィル)。ハッチはグラウンドに接続する必要があります。

センサーとシールド電極間の推奨エアギャップは 1mm です。

7.4.12.2 ガード センサーのレイアウトガイドライン

ガード センサー セクションで説明したように、ガード センサーは **Figure 149** に示すようにすべてのセンサーを取り囲んでいる銅配線です。

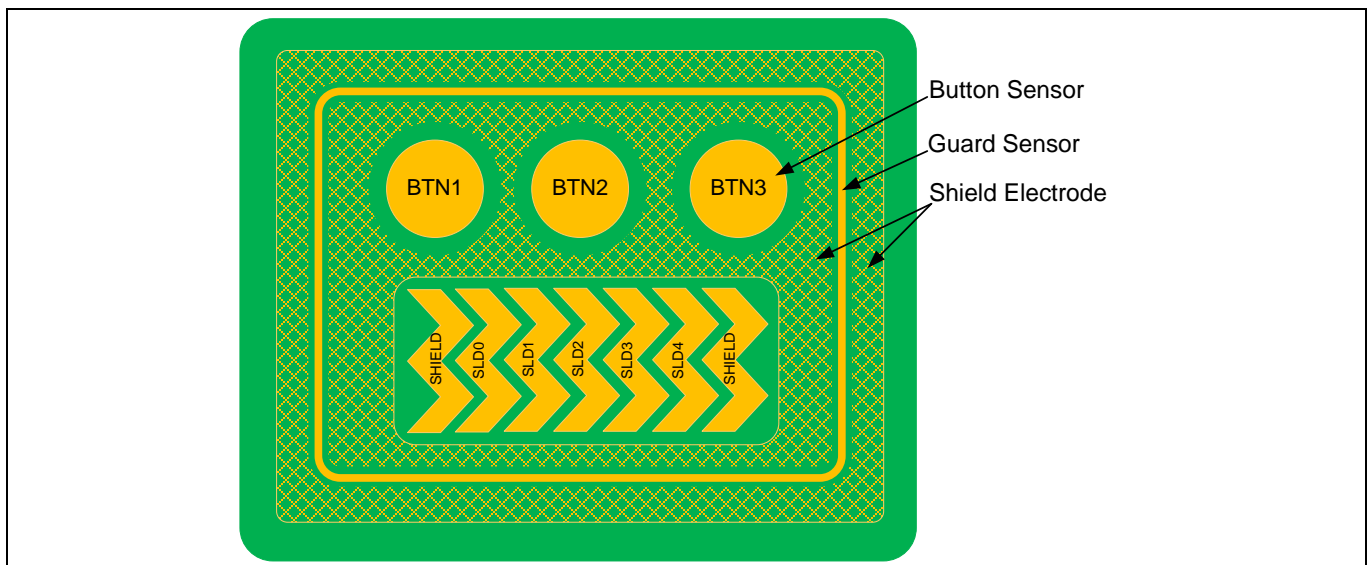


Figure 149 シールド電極とガード センサーを実装した PCB レイアウト

ガード センサーは、タッチ面上に液体の流れが存在するときのみ、トリガーする必要があります。ガード センサーが液滴の存在でオンにならないようにシールド電極パターンがガード センサーを囲むことを確認してください。ガード センサーは、以下の条件を満たすように配置する必要があります。

- ガード センサーは、タッチ面上に液体の流れが存在するとき、最初にオンになるセンサーとなる必要があります。これを実現するためには、通常、ガード センサーはすべてのセンサーを囲むように配置されます。
- ガード センサーは、ボタンやスライダー センサーを押しているとき、誤ってタッチされないようにしてください。そうでない場合、ガード センサーがオフになるまでは、ボタン センサーやスライダ

設計上の注意事項

センサーのスキャンが無効になり、CAPSENSE™システムが非稼動になります。ガードセンサーが誤ってトリガーされないようにするために、ガードセンサーをセンサーから 1cm 以上離れているところに配置します。

ガードセンサーを実装するためには、以下のガイドラインに従ってください。

- ガードセンサーは、角が丸い長方形であり、すべてのセンサーを取り囲む必要があります。
- ガードセンサーの推奨厚さは 2mm です。
- ガードセンサーとシールド電極間の推奨間隔は 1mm です。

PCB にガードセンサーを実装するための領域がない場合、ガードセンサーの機能は、ファームウェアに実装できます。例えば、さまざまなセンサーの ON/OFF ステータスを使用して、ユースケースデータに応じて液体の流れを検出できます。

以下の条件は、タッチ面上の液体の流れを検出するために使用できます。

- 液体の流れがあるとき、2 個以上のボタンセンサーが 1 度にアクティブになります。設計がマルチタッチセンシングを必要としない場合、誤ったトリガーを防ぐために、これを検出してすべてのボタンセンサーのステータスが拒否できます。
- スライダーでは、オンになっているスライダーセグメントが隣接するセグメントでない場合、スライダーセグメントのステータスをリセットする、または計算されたスライダーの重心値を拒否できます。
- ユースケースデータから水滴による誤タッチを検出するファームウェアアルゴリズムを作成して、誤タッチ拒否機能センサーを改善できます。

7.4.12.3 グランドリングによる耐液性

Figure 150 に示すように、複数のアプリケーションで、向上した ESD および EMI/EMC 性能のために、基板のペリフェラルを囲むグランドリング (固体配線またはハッチ フィル) が必要となります。基板を囲むグランドリングがあると、液滴がセンサーとグランドセンサーとの間にあるとき、センサーが誤ってトリガーすることになります。したがって、最上層にグランドリングを配置しないことを推奨します。設計で最上層にグランドリングが必要となる場合、最小厚さ (8 mil) のグランドリングを使用してください。

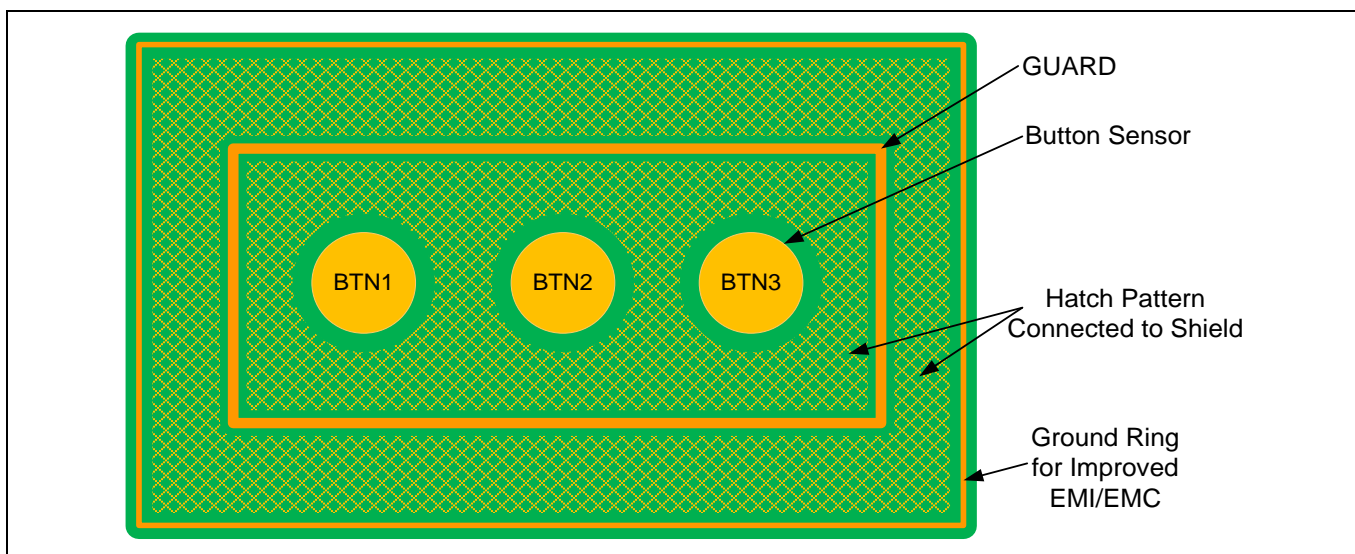


Figure 150 向上した ESD および EMI/EMC 性能用のグランドリング付き CAPSENSE™ 設計

設計上の注意事項

7.4.13 回路図のルール チェックリスト

Table 34 に、CAPSENSE™回路図を検証するためのチェックリストを示します。

Table 34 回路図のルール チェックリスト

No.	カテゴリ	推奨事項/備考
1	C _{MOD}	2.2nF ピンの選択は Table 35 を参照してください。
2	C _{SH_TANK}	シールド電極を使用する場合は 10nF。その他、該当なし。シールド電極と C _{SH_TANK} の使用の詳細は、それぞれ被駆動シールド信号およびシールド電極および CAPSENSE™ CSD シールドを参照してください。 ピンの選択は Table 35 を参照してください。
3	C _{INTA} /C _{INTB}	470pF。ピンの選択は Table 35 を参照してください。
3	入力ラインの直列抵抗	自己容量では 560Ω、相互容量では 2kΩ。詳細は CAPSENSE™ピンの直列抵抗を参照してください。
4	センサーピンの選択	可能な場合、スイッチング/通信信号を持つ GPIO の近くに配置するピンは避けてください。可能な限り、DC 負荷 (LED および I ² C ピン) は CAPSENSE™ピンからフルポートで物理的に切り離す必要があります。詳細はセンサーピンの選択を参照してください。
5	GPIO ソース/シンク電流	CAPSENSE™ブロックがセンサーをスキャンしている間に GPIO を通る総シンク電流が 40mA 以下であることを確認してください。

7.4.13.1 外部コンデンサ接続のピン選択

CAPSENSE™の基礎セクションで説明したとおり、CAPSENSE™には、信頼性の高い動作のために、外部コンデンサ (C_{MOD} (CSD センシング方式)、C_{TANK} (シールドが実装された場合のみ)、および C_{INTX} (CSX センシング方式)) が必要です。PSoC™ Creator 3.3 SP2 以降、C_{MOD} および C_{SH_TANK} に対応するピンの数は増え、設計の柔軟性が向上します。Table 35 に PSoC™ Creator 3.3 SP2 またはそれ以降のバージョンの C_{MOD}、C_{INTX} および C_{SH_TANK} コンデンサ用に推奨されるピンの一覧を示します。

Note: PSoC™ 4100/PSoC™ 4200 では、P4[2]以外のピンが C_{MOD} 用に選択される場合、推奨された C_{MOD} ピンは他の機能が使用不可となります。例えば、PSoC™ Creator で PSoC™ 4200 デバイスに C_{MOD} を P2[0] に配線すると、P2[0] と P4[2] の両方が使用可能です。

Table 35 外部コンデンサ用に推奨するピン

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™ 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]
PSoC™ 6 MCU	P7[1]	P7[2]
PSoC™ 4S シリーズ, 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]

設計上の注意事項

Table 36 外部コンデンサに対応するピン

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})	C _{INTA}	C _{INTB}
PSoC™ 4000	Port0[0:7], Port1 [0:7] P2[0]	Port0 [0:7], Port1 [0:7] P2[0]	P0[4]	P0[2]
PSoC™ 4100	Port0 [0:7], Port1 [0:7] Port2 [0:7], Port3 [0:7] P4[2]	Port0 [0:7], Port1 [0:7] Port2 [0:7], Port3 [0:7] P4[3]	非対応	非対応
PSoC™ 4200	Port0 [0:7], Port1 [0:7] Port2 [0:7], Port3 [0:7] P4[2]	Port0 [0:7], Port1 [0:7], Port2 [0:7], Port3 [0:7] P4[3]	Port0 [0:7], Port1 [0:7] Port2 [0:7], Port3 [0:7]	Port0 [0:7], Port1 [0:7] Port2 [0:7], Port3 [0:7]
PSoC™ 4200M	CSD0: Port0 [0:7], Port1 [0:7] Port2 [0:7], Port3 [0:7] Port4 [0:6], Port6 [0:5] Port7 [0:1]	CSD0: Port0 [0:7], Port1 [0:7] Port2 [0:7], Port3 [0:7], Port4 [0:6], Port6 [0:5], Port7 [0:1]	CSD0: P4[2]	CSD0: P4[3]
	CSD1: 非対応	CSD1: 非対応	CSD1: 非対応	CSD1: 非対応
PSoC™ 4200L	CSD0: Port0 [0:7], Port1 [0:7] Port2 [0:7], Port3 [0:7] Port4 [0:6], Port6 [0:5] Port7 [0:7], Port10 [0:7], Port11 [0:7]	CSD0: Port0 [0:7], Port1 [0:7] Port2 [0:7], Port3 [0:7] Port4 [0:6], Port6 [0:5] Port7 [0:7], Port10 [0:7], Port11 [0:7]	CSD0: P4[2]	CSD0: P4[3]
	CSD1: Port5 [0:7], Port8 [0:7] Port9 [0:7]	CSD1: Port5 [0:7], Port8 [0:7] Port9 [0:7]	CSD1: P5[0]	CSD1: P5[1]
PSoC™ 4 Bluetooth® LE	Port0 [0:7], Port1 [0:7] Port2 [0:7], Port3 [0:7] Port4 [0:1], Port5 [0:1] Port6 [0:1]	Port0 [0:7], Port1 [0:7] Port2 [0:7], Port3 [0:7] Port4 [0:1], Port5 [0:1] Port6 [0:1]	P4[0]	P4[1]
PSoC™ 6 MCU	P7[1] または P7[2] または P7[7]	P7[1] または P7[2] または P7[7]	P7[1]	P7[2]
PSoC™ 4S シリーズ, PSoC™ 4100S Plus	P4[2], P4[3], P4[1]	P4[2], P4[3], P4[1]	P4[2]	P4[3]
PSoC™ 4100PS	P5[0], P5[2], P5[3]	P5[0], P5[2], P5[3]	P5[2]	P5[3]
PSoC™ 4100S Max	Channel0: P4[0], P4[2]	Channel0: P4[1], P4[3]	該当なし	該当なし
	Channel1: P7[0], P5[1]	Channel1: P7[1], P5[2]		

7.4.13.2 センサーピンの選択

CAPSENSE™の基礎で説明されるように、PSoC™は CSD および CSX 静電容量センシング方式をサポートします。各 CSD センサーには 1 つのセンサーピンが必要であり、CSX センサーには、各センシング技術に必要な外部コンデンサに加えて、Tx および Rx 電極用に 2 つのセンサーピンが必要です。

設計上の注意事項

センサーピンの選択は、CAPSENSE™ センサートレースと通信またはその他のトグル GPIO トレースが適切なポート/ピン割り当てによって分離されるような方法で行う必要があります。以下は、いくつかの推奨ガイドラインです。

- PWM、I2C 通信ライン、LED などのスイッチング信号をセンサーおよびセンサートレースから分離してください。それらを少なくとも 4mm 離して配置し、クロストークを回避するために、CAPSENSE™ トレースとスイッチング信号の間のハッチングされたグラウンドを埋めてください。
- CAPSENSE™ のノイズを低減するために、さまざまなポートに DC 負荷の配置を分散してください。デジタル I/O を単一のポートに集中させるのではなく、異なるポートに分散させることを推奨します。
- CAPSENSE™ ブロックがセンサーをスキャンしている間、CAPSENSE™ ブロックがセンサーをスキャンしている間、GPIO を流れるソースまたはシンクの合計電流を 40mA 未満に制限してください。CAPSENSE™ センサーのスキャン中に 40mA を超える電流をシンクすると、センサーの raw カウントに過度のノイズが発生する可能性があります。
- PSoC™ 4 デバイスの場合、LED、I2C/UART 通信ピンなどのすべてのデジタル DC 負荷を VSSD のみで給電されるポートに配置することを推奨します。VSSD から電力が供給されるポートの決定については、[デバイスデータシート](#)を参照してください。VSSA を搭載したポートに DC 負荷をかけると、VSSA が上にシフトします。CAPSENSE™ は VSSA を搭載しているため、パフォーマンスに影響します。
- PSoC™ 6 ファミリのデバイスの場合は以下のとおりです。
 - [Table 37](#) に、CAPSENSE™ をサポートするポートを示します。CAPSENSE™ にポート 5, 6, 7, および 8 を選択すると、ノイズが少なくなります。
 - LED、I2C、UART、SPI、SMIF 通信ピンなどのすべてのデジタルスイッチングピンを、CAPSENSE™ ポートと共有されない別の電源ドメインから電力が供給されるポートに配置することを推奨します。[Table 38](#) に、ポート、それらの供給ドメイン、およびこれらのポートを CAPSENSE™ で使用するための推奨事項を示します。詳細については、[デバイスデータシート](#)のエラッタセクションを参照してください。これらのガイドラインから逸脱すると、raw カウントのレベルシフトによりノイズが発生する可能性があります。詳細については、[GPIO を切り替えたとき raw カウントがレベルシフトまたはノイズの増加を示す](#)、を参照してください。供給ドメインをさらに分離するためには、[Figure 152](#) に示すように、フェライトビーズを使用して外部から分離することを推奨します。

Table 37 PSoC™ 6 デバイスの CAPSENSE™ 対応ポート

デバイス	CAPSENSE™ 対応ポート
CY8C62x6 , CY8C62x7	P0, P1, P2, P4, P5, P6, P7, P8, P9, P10, P11
CY8C63x6 , CY8C63x7	P0, P1, P2, P4, P5, P6, P7, P8, P9, P10, P11
CY8C62x5	P7.0~P7.7, P8.0~P8.3, P9.0~P9.3

Table 38 PSoC™ 6 デバイスでの CAPSENSE™ のポート使用についての推奨事項

ポート	供給ドメイン	CAPSENSE™ に推奨	CAPSENSE™ を使用した通信、LED、およびその他の高周波機能に使用する場合の GPIO の推奨事項
P0	V _{BACKUP}	無し ¹⁷	スイッチング周波数 < 8MHz
P1	V _{DDD}	無し ¹⁷	スイッチング周波数 < 1MHz, SLOW スルーレート
P2, P3, P4	V _{DDIO2}	無し ¹⁷	スイッチング周波数 < 25MHz

¹⁷ 追加の CAPSENSE™ ピンが必要で、ポート P1, P9, および P10 の GPIO を CSX センサーの Tx 電極として使用する必要がある場合は、Tx クロック周波数を 1 MHz 以内に制限し、SLOW スルーレートを使用します。[Figure 151](#) に、ModusToolbox™ プロジェクトの [Device configurator](#) を使用して GPIO のスルーレートを選択する方法の例を示します。CAPSENSE™ の推奨ポート以外のポートを使用すると、rawcount でより高いノイズが発生する可能性があることに注意してください。

設計上の注意事項

ポート	供給ドメイン	CAPSENSE™に推奨	CAPSENSE™を使用した通信, LED, およびその他の高周波機能に使用する場合の GPIO の推奨事項
P5, P6, P7, P8	V _{DDIO1}	有り	推奨しません
P9, P10	V _{DDIOA}	無し ¹⁷	スイッチング周波数< 1MHz, SLOW スルーレート
P11, P12, P13	V _{DDIO0}	無し ¹⁷	スイッチング周波数< 80MHz
P14	V _{DDUSB}	無し ¹⁷	該当なし

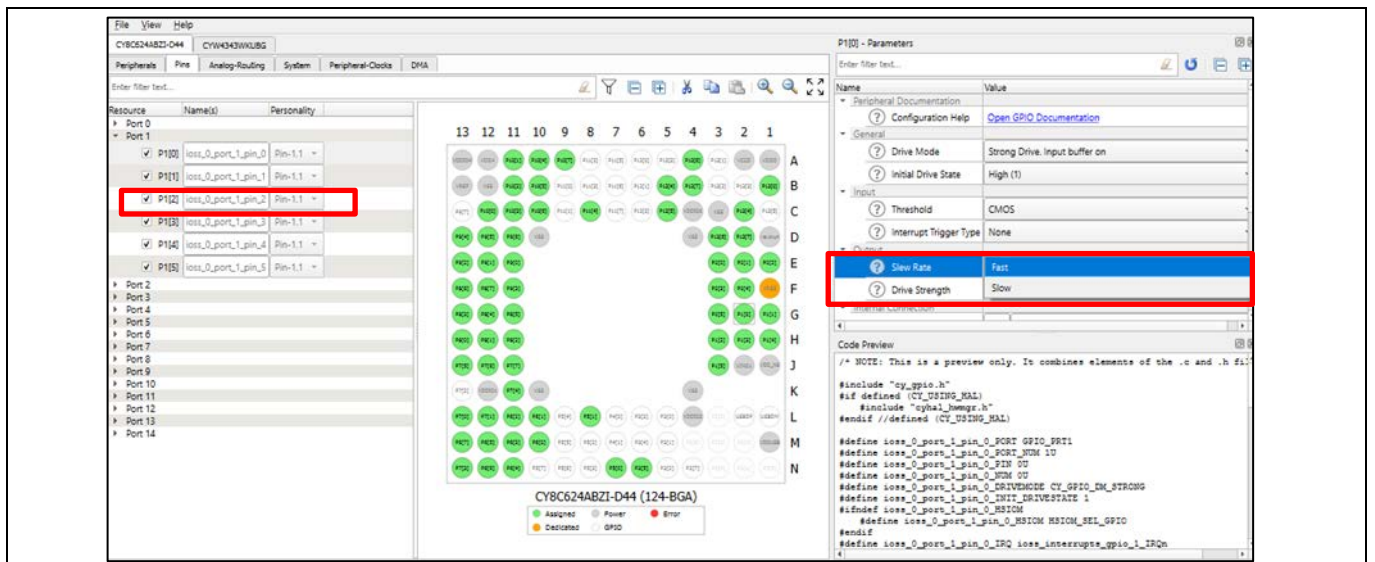


Figure 151 GPIO のスルーレートの選択

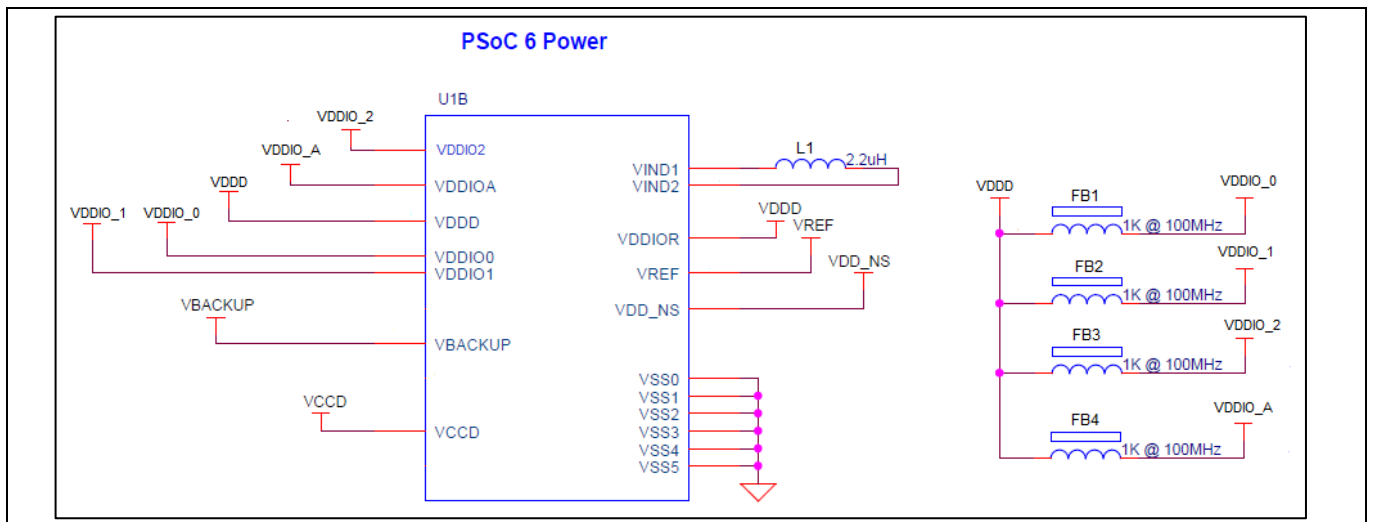


Figure 152 外部で分離された供給ドメイン

7.4.14 レイアウトルールチェックリスト

Table 39 に目的のレイアウト設計の検証を支援するチェックリストを示します。

Table 39 レイアウトルールチェックリスト

設計上の注意事項

No.	区分	最小値	最大値	推奨事項/備考	
1	ボタン	形状	該当なし	円形または角が丸い長方形です。	
		サイズ	5mm	15mm	10mm
		グランドハッチまでの間隔	0.5mm	2mm	オーバーレイの厚さに等しくなければなりません。
2	スライダー	セグメント幅	1.5mm	8mm	8mm
		セグメント間隔	0.5mm	2mm	0.5mm
		セグメントの高さ	7mm	15mm	12mm
3	オーバーレイ	タイプ	該当なし	該当なし	高い比誘電率の材料(導体を除く)。 センサー基板とオーバーレイ/ケーシングのフロントパネル間のエアギャップを取り除きます。
		ボタンの厚さ	該当なし	5mm	
		スライダーの厚さ	該当なし	5mm	
		タッチパッドの厚さ	該当なし	0.5mm	
4	センサー配線	幅	該当なし	7mil	使用されている PCB 技術で可能な限りの最小幅を使用します。
		長さ	該当なし	標準 (FR4) PCB の場合: 300mm フレックス PCB の場合: 50mm	できるだけ短くします。
		グランドおよび他の配線までの間隔	0.25mm	該当なし	配線をできるだけ短くし、間隔を最大限に維持します。
		配線	該当なし	該当なし	センサーレイヤの反対面に配線し、他の配線から分離します。CAPSENSE™ではない任意の配線が CAPSENSE™配線と交差する場合、交差部分が直交であるようにします。急カーブを避けてください。

設計上の注意事項

No.	区分		最小値	最大値	推奨事項/備考
5	ビア	ビア数	1	2	センサーレイヤの反対面の配線に少なくとも 1 個のビアが必要です。
		穴のサイズ	該当なし	該当なし	10mil
6	グラウンド	ハッチフィルの割合	該当なし	該当なし	寄生容量を削減するためにハッチグラウンドを使用します。典型的なハッチングは次のとおり。 最上層: 25% (7mil ライン、45mil 間隔)。 最下層: 17% (7mil ライン、70mil 間隔)。
7	直列抵抗	配置	該当なし	該当なし	抵抗を PSoC™ピンの 10mm 以内に配置します。基板上の直列抵抗の配置例は Figure 153 を参照してください。
8	シールド電極	広さ	該当なし	1cm	PCB 面に余裕がある場合、1cm の広さを使用します。
9	ガードセンサー (耐水性)	形状	該当なし	該当なし	角が丸い長方形です。
		厚さ	該当なし	該当なし	ガード配線の推奨厚さは 2mm で、ガード配線からシールド電極までの推奨距離は 1mm です。
10	C _{MOD}	配置	該当なし	該当なし	PSoC™ピンの近くに配置します。PCB 上の C _{MOD} の配置例は Figure 153 を参照してください。
11	C _{SH_TANK}	配置	該当なし	該当なし	PSoC™ピンの近くに配置します。基板上の C _{SH_TANK} の配置例は Figure 153 を参照してください。
12	C _{INTA}	配置	該当なし	該当なし	PSoC™ピンの近くに配置します。PCB 上の C _{INTA} の配置例は Figure 153 を参照してください。
13	C _{INTB}	配置	該当なし	該当なし	PSoC™ピンの近くに配置します。PCB 上の C _{INTA} の配置例は Figure 153 を参照してください。

設計上の注意事項

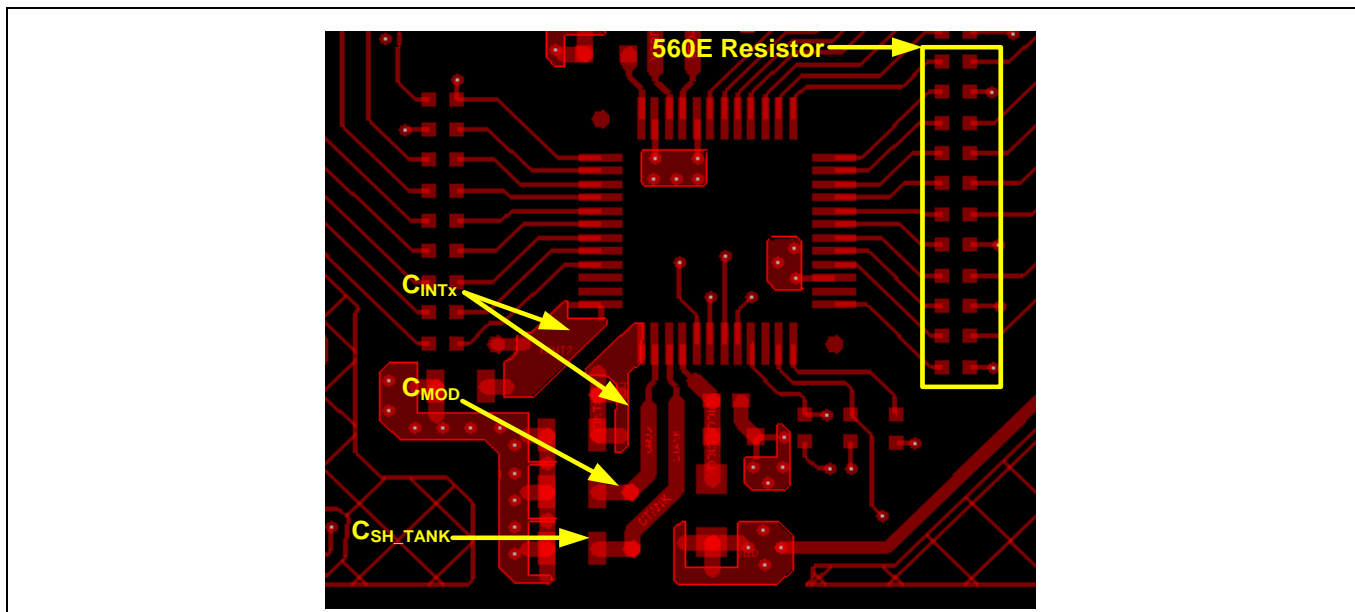


Figure 153 PSoC™ 4200M デバイスにおける入力ライン上の C_{MOD} , C_{INTx} , C_{SH_TANK} および直列抵抗の配置例

7.5 CAPSENSE™ システムのノイズ

7.5.1 Finger Injected ノイズ

システムの電源設計が不十分な場合、デバイスの電源と接地電源は、コモンモード方式でフィンガーグラウンド (アース接地) に対して電圧が変動します。このタイプのノイズは、コモンモードノイズと呼ばれます。Figure 154 に、電源の 5V と 0V の両方の出力リードが互いに 5V のままですが、「コモンモード」方式で一緒に上下に移動するコモンモードノイズを示します。

ボタンに指で触れるまで、これは問題ではありません。ボタンを指でタッチすると、同じアースに (容量性の) パスが導入され、電荷の流れのパスが作成されます。これは、指のタッチ位置に正確に注入されたノイズ信号に相当します。電源のコモンモードノイズによって引き起こされるこの注入ノイズは、Finger Injected ノイズと呼ばれます。AC 電源アプリケーションではボタンを指で触れている間のみ観察され、バッテリー電源アプリケーションでは発生しません。

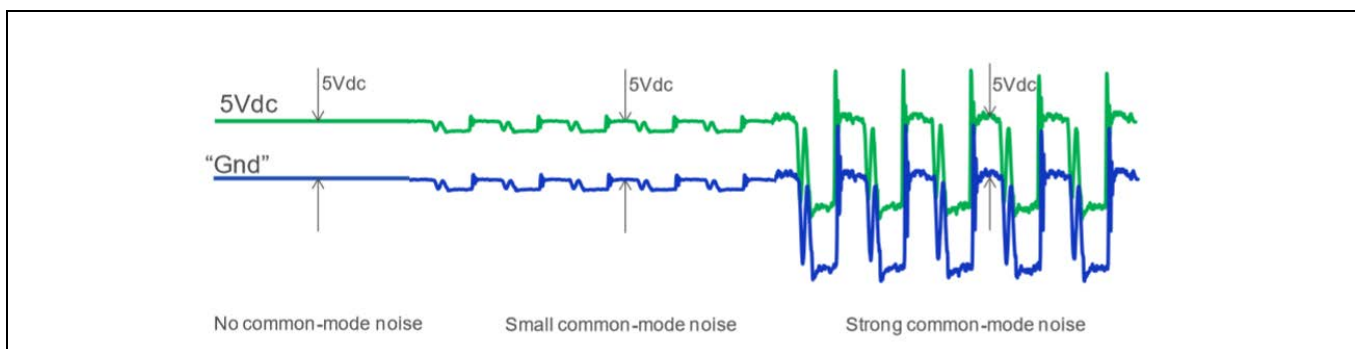


Figure 154 電源のコモンモードノイズ

設計上の注意事項

AC 電源で給電されるシステム全体をユーザーの手に持つと、システム全体が十分にアースに接地され、触れる指からアースに重大な「コモンモード」ノイズが流れることはありません。ただし、システムが電源に接続されて机の上に置かれている場合は、ボタンに触れると、問題のあるアースへの放電経路が発生する可能性があります。

7.5.1.1 Finger injected ノイズを減らすための推奨事項

ここで説明されているレイアウトと回路図のガイドラインに適切に従うことで、Finger injected ノイズを減らせます。Finger injected ノイズを減らすための一般的な推奨事項を以下に説明します。

- a) ボタンの周りの PCB ボードをハッチングパターンで埋め、デバイスのアースに接続します。グラウンド面のセクションに記載される推奨事項に従ってください。
Figure 155 に、相互容量ボタンの指で注入されたノイズに対するグラウンドの影響を示します。これは、CSD センシング技術にも当てはまります。左の図では、システムのボタンの周りにハッチングされたグラウンドがなく、指から注入されたノイズのほとんどは、指と Rx 電極の間に形成された静電容量を介してデバイスの Rx ピンに渡されます。右の図では、システムのボタンの周りにハッチングされたグラウンドがあるため、指で注入されたノイズには代替の流れ経路があり、デバイスの Rx ピンに到達するノイズが減少します。

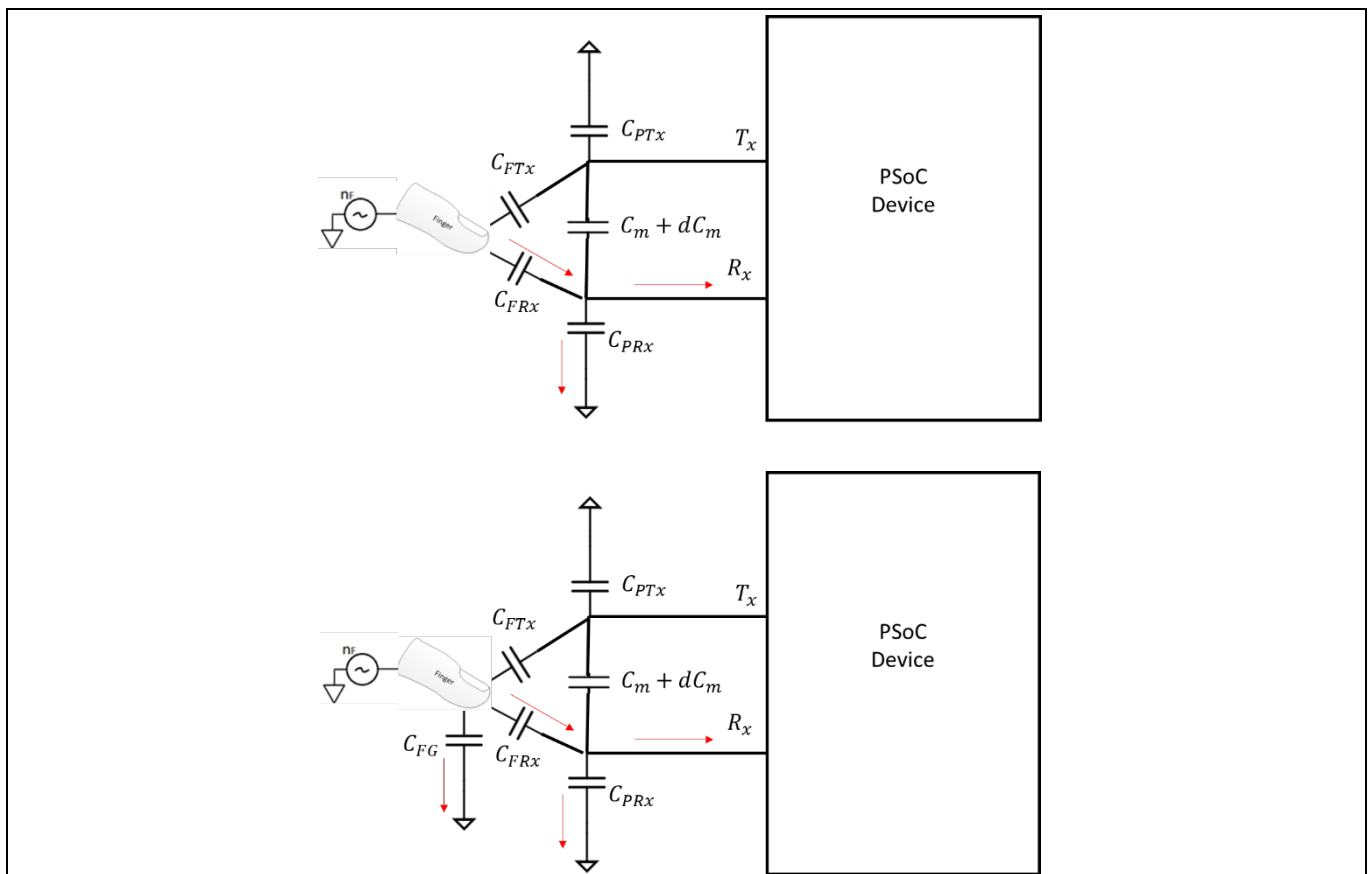


Figure 155 Finger injected ノイズに対する接地の影響

- b) システムのより良い電源設計は、コモンモードノイズを簡単に排除でき、それは次に Finger injected ノイズを減らすでしょう。
- c) CAPSENSE™ コンポーネントで利用可能なソフトウェア技術を使用して、最適なセンシングクロック周波数の選択やマルチ周波数スキャンなど、Finger injected ノイズに対抗します。

設計上の注意事項

- d) オーバーレイの厚さを増やすと、指と Rx 電極の間に形成される静電容量が減少するため、Finger injected ノイズが減少します。

7.5.2 VDDA ノイズ

VDD 電源の不要な電圧リップルによるシステムのノイズは、VDDA ノイズと呼ばれます。

7.5.2.1 VDDA ノイズを低減するための推奨事項

この章のレイアウトと回路図のガイドラインに正しく従うことで、VDDA ノイズを減らせます。VDDA ノイズを低減するための一般的な推奨事項は次のとおりです。

- クリーンな電源を使用し、VDD リップルがデバイスのデータシートに記載されている制限を下回ってください。
- VDD 電源ラインでフィルターまたは LDO レギュレーターを使用してください。
- 電源ピンにデカップリングコンデンサを使用して、電源からの伝導ノイズを低減してください。
- 高周波ノイズを低減するには、電源または通信回線の周囲にフェライトビーズを配置してください。
- デバイスデータシート**の電源セクションに記載されている適切な電源構成を選択し、デバイスの内部レギュレーターを使用すると、VDDA ノイズの低減に役立つ場合があります。

7.5.3 外部ノイズ

ESD、EMI、伝導ノイズなどのルーティングトレースラインを介してシステムに注入されるノイズは、外部ノイズのカテゴリに分類されます。ここでは、外部ノイズの影響を減らすための推奨ガイドラインについて説明します。

7.5.3.1 ESD 保護

CAPSENSE™で使用される非導電性オーバーレイ材料は、ESD に対する固有の保護を提供します。**Table 40** に、CAPSENSE™センサーを 12 kV 放電から保護するために必要なさまざまなオーバーレイ材料の厚さを示します (IEC 61000-4-2 仕様に準拠)。

Table 40 ESD 保護のためのオーバーレイの厚さ

材料	ブレイクダウン電圧 (V/mm)	12 kV ESD から保護するための最小オーバーレイ厚さ ESD (mm)
空気	1200 - 2800	10
木 - 乾燥したもの	3900	3
ガラス - 一般のもの	7900	1.5
ガラス - ホウケイ酸 (Pyrex®)	13,000	0.9
PMMA プラスチック (Plexiglas®)	13,000	0.9
ABS	16,000	0.8
ポリカーボネート (Lexan®)	16,000	0.8
Formica	18,000	0.7
FR-4	28,000	0.4
PET フィルム (Mylar®)	280,000	0.04
ポリイミドフィルム (Kapton®)	290,000	0.04

設計上の注意事項

オーバーレイの素材で十分に保護できない (例えば、他の方向からの ESD に対する保護) 場合、**防止, リダイレクト, ESD 保護デバイス**の順に従って、他の ESD 対策を適用できます。

7.5.3.1.1 ESD 放電の防止

ESD 放電が PSoC™に届かないようにすることが一番の対策です。PSoC™へのすべてのパスは、機器の表面で可能な最大 ESD 電圧よりも高い耐電圧があることを確認してください。予想される ESD ソースと PSoC™間の適切な距離も維持する必要があります。**Figure 156**にある例では、L1 および L2 が 10mm 以上である場合、システムは 12kV ESD に耐えられます。

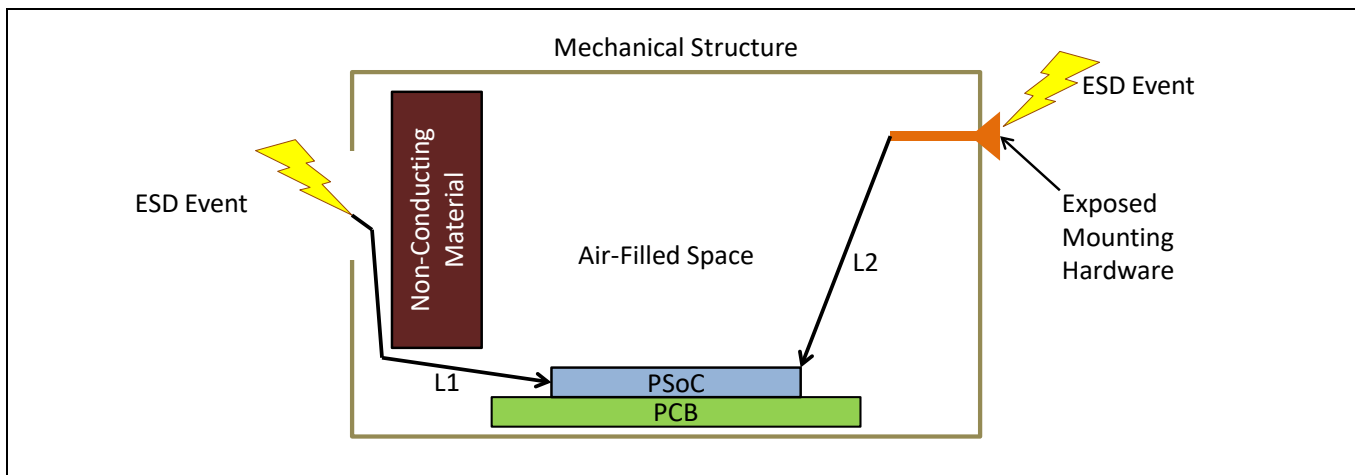


Figure 156 ESD パス

適切な距離を保つことが不可能な場合には、高絶縁破壊電圧を有する非導電材料の保護レイヤを予想される ESD ソースと PSoC™の間に設けてください。厚さ 5mil の Kapton®テープは 18kV に耐えられます。他の素材の絶縁耐力は **Table 40** を参照してください。

7.5.3.1.2 リダイレクト

製品がぎっしりと詰まっている場合は、放電現象を避けることは難しいかもしれません。このような場合、ESD を逃がすことにより、PSoC™を ESD から保護できます。一般的には、**Figure 157** で示すように、回路基板の周辺にグラウンドリングを配置します。グラウンドリングはシャーシグラウンドに接続する必要があります。また、ボタンやスライドセンサーの周囲にハッチンググラウンド面を施すと、センサーおよび PSoC™への ESD の影響も回避できます。

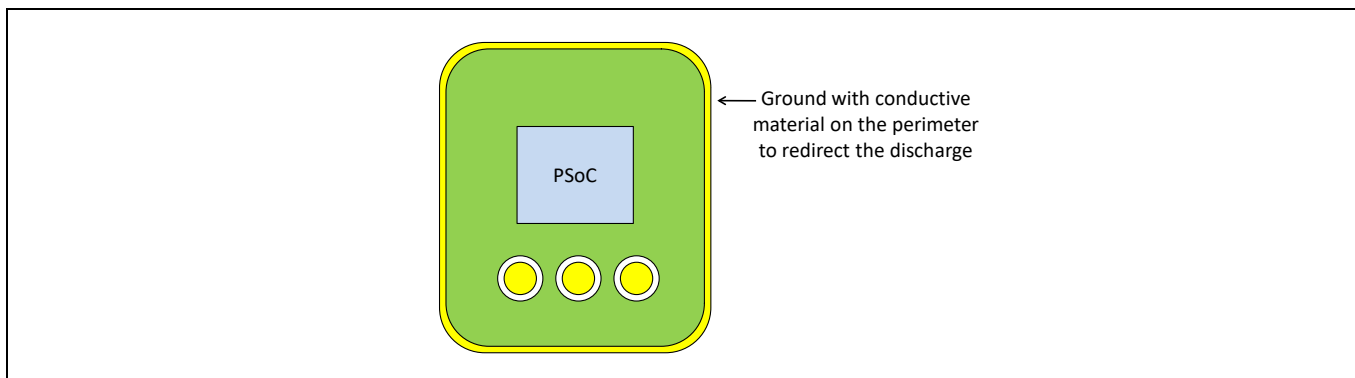


Figure 157 グラウンド リング

設計上の注意事項

7.5.3.1.3 ESD 保護デバイス

ESD 保護用デバイスが ESD に脆弱な配線に使用できます。CAPSENSE™ 感度が低下しないように低入力静電容量の ESD 保護デバイスを選択します。推奨される ESD 保護デバイスを **Table 41** に示します。

Table 41 ESD 保護デバイス

ESD 保護デバイス		入力静電容量	リーク電流	コンタクト放電の 最大 ESD 耐圧	空中放電の 最大 ESD 耐圧
メーカー	型番				
Littelfuse	SP723	5 pF	2 nA	8 kV	15 kV
Vishay	VBUS05L1-DD1	0.3 pF	0.1 μA	±15 kV	±16 kV
NXP	NUP1301	0.75 pF	30 nA	8 kV	15 kV

7.5.3.2 電磁適合性 (EMC) に対する考慮事項

EMC は、電子システムの動作に悪影響を与える可能性がある、電磁エネルギーの発生、伝達、および受信に関連します。多くのエレクトロニクスデバイスは、放射エネルギーに対応し、外部のイベントの影響を受けない耐性を持つ必要があります。世界のいくつかの標準化団体が、地域規制を設定し、エレクトロニクスデバイスが互いに干渉しないようにしています。

CMOS アナログおよびデジタル回路の入力インピーダンスは非常に高いです。その結果、外部電界に敏感です。そのため、放射または導電性ノイズがあるところで正常動作を確保するためには、適切な注意を払う必要があります。

コンピューティング デバイスは、米国では、意図的でない放熱器用に FCC の第 15 章の項目 B により規定されます。ヨーロッパと世界の他の国々の基準は CENELEC から適応されます。それらは、妨害波について CISPR 標準 (ENxxxx 標準としての二重標識) で、耐性と安全性について IEC 標準 (ENxxxx 標準としても二重標識) で保護されます。

コンピューティング デバイスの場合、一般妨害波仕様は EN55022 です。この標準は放射と伝導妨害を対象にします。米国での医療機器は FCC ではく、医療機器用の欧州基準の EN55011 の条件を含む FDA により規定されます。モーター制御を含むデバイスは EN55014、照明装置は EN50015 に従います。

基本的には、これらの仕様は放射と伝導妨害に関わる同様な性能の制限を持ちます。放射と伝導耐性 (感受性) の性能要件は EN61000-4 の複数の項で指定されます。この標準はライン電圧過渡現象、ESD および複数の安全問題も説明します。

7.5.3.2.1 放射干渉およびエミッション

PSoC™ 4 および PSoC™ 6 Bluetooth® LE は堅牢な CAPSENSE™ 性能を提供しますが、放射電気エネルギーはシステム測定に影響を与え、CAPSENSE™ プロセッサ コアの動作に潜在的に影響を与える場合があります。干渉は PCB レベルでセンサー トレースや他のデジタルおよびアナログ入力を経て CAPSENSE™ デバイスに入り込みます。CAPSENSE™ デバイスは、電磁環境適合性 (EMC) の問題対応に放射放出の形で寄与できます。

以下の技術を使用して放射干渉および放出を最小限にしてください。

ハードウェア設計上の注意事項

グラウンド面

一般的には、PCB の適切なグラウンド面は RF 放射と干渉の両方を低減します。しかし、CAPSENSE™ センサーの近くのベタグラウンド面またはこれらのセンサーを PSoC™ のピンに接続する配線は、センサーの寄

設計上の注意事項

生容量を増加させます。したがって、**PCB レイアウトガイドライン**の**グランド面**で説明したように、センサー周囲およびセンサーの下のプリント基板の最下層でハッチンググランド面を使用することを推奨します。CAPSENSE™センサーと配線から遠く離れた PCB のデバイスおよび他の回路の下にベタグランド面が使用できます。CAPSENSE™センサーまたは配線から 1cm 以内にベタグランドパターンの使用は推奨されません。

CAPSENSE™ピンの直列抵抗

すべての CAPSENSE™コントローラーピンは、関連する寄生容量 (C_p) を持ちます。**Figure 158** に示すように、外部抵抗の追加はローパス RC フィルターを形成し、ピンに伴う RF ノイズ振幅を減衰させます。また、この抵抗は CAPSENSE™センサーの寄生容量とともに RF 放射を大幅に削減するローパスフィルターを形成します。

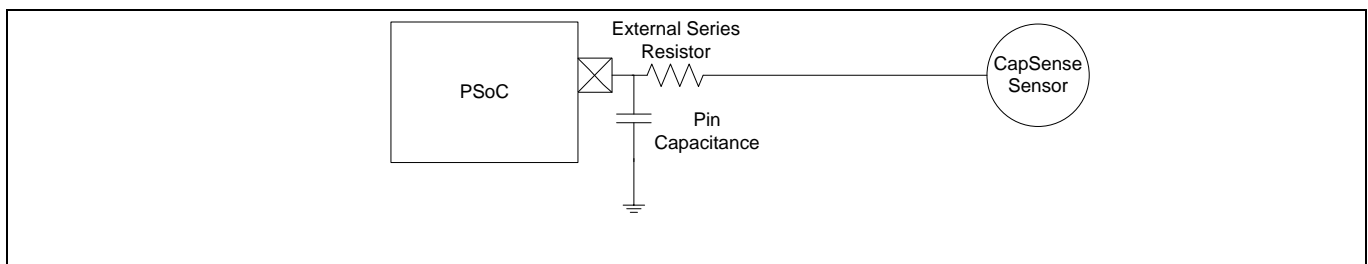


Figure 158 RC フィルター

配線によって拾われた放射ノイズがデバイスの入力でフィルター処理されるように、直列抵抗をデバイスのピンの近くに配置する必要があります。したがって、ピンから 10mm 以内で直列抵抗を配置することを推奨します。

PCB に銅を使用する CAPSENSE™設計では、CAPSENSE™入力ラインの推奨抵抗値は 560Ω です。この抵抗を追加すると、 C_p を等価抵抗値に変換するスイッチトキャパシタ回路のとき定数が増加します (**GPIO セルの静電容量-電流変換器**を参照してください)。直列抵抗の値が 560Ω より大きい場合、スイッチング回路の時定数がより遅くなるため、放射と干渉が抑制されますが、転送可能な電荷量は制限されます。これにより信号レベルが下がり、そして **SNR** 比が下がります。小さい値の場合は SNR に優れていますが、RF へのブロック効果は低下します。

デジタル通信ライン上の直列抵抗

I²C や SPI などの通信ラインも、直列抵抗の恩恵を受けて、330Ω は通信ラインにおける直列抵抗に推奨されます。通信ラインは長い配線で、CAPSENSE™配線のアンテナとして作用します。I²C 通信ラインの推奨プルアップ抵抗値は 4.7kΩ です。これらのライン上に 330Ω より大きい直列抵抗を配置すれば、 V_{IL} および V_{IH} 電圧レベルは仕様範囲外になります。330Ω の直列抵抗なら、PSoC™が LOW 信号を出力すれば、 V_{IL} レベルが $0.3V_{DD}$ の I²C 仕様範囲内に維持されるため、I²C の動作には影響を与えません。

設計上の注意事項

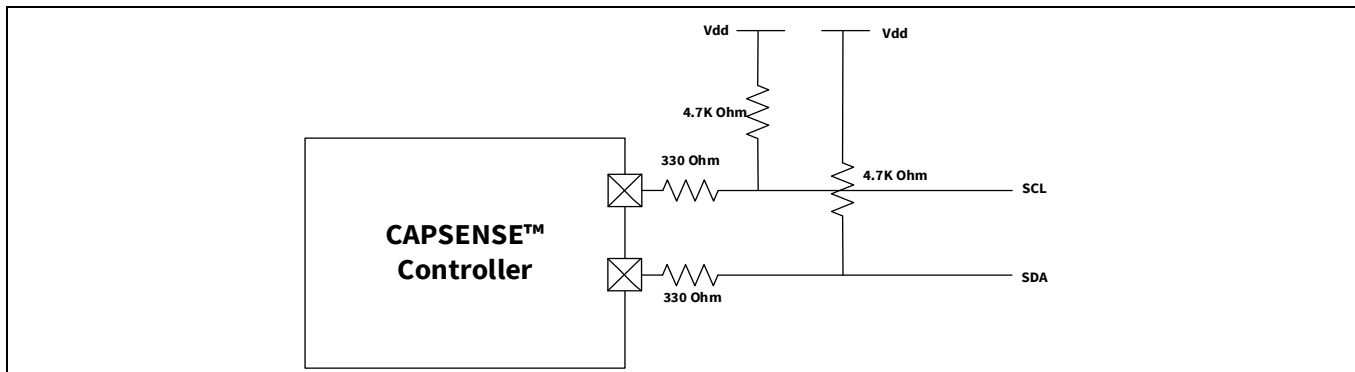


Figure 159 通信ライン上の直列抵抗

配線長

長い配線は短い配線より多くのノイズを拾います。長い配線は C_p も増加させます。そのため、可能な限り配線を短くします。

電流ループ領域

レイアウトときの重要な注意点の1つは、電流のリターン経路を最短にすることです。電流がループ内に流れるため、これは重要です。高速信号のための適切なリターン経路がない限り、リターン電流は、より長いリターン経路を通して流れ、より大きなループを形成します。これに起因して放射と干渉が増える場合があります。

Figure 160 に示すように、CAPSENSE™ グランドハッチとデバイス周辺のグランドフィルを分離すると、センサーのスイッチング電流は長いリターン経路を使用することがあります。CAPSENSE™ センサーが高い周波数で切り替えられるため、リターン電流は重大な EMC 問題を引き起こすことがあります。したがって、Figure 161 に示されるように単一のグランドハッチを使用する必要があります。

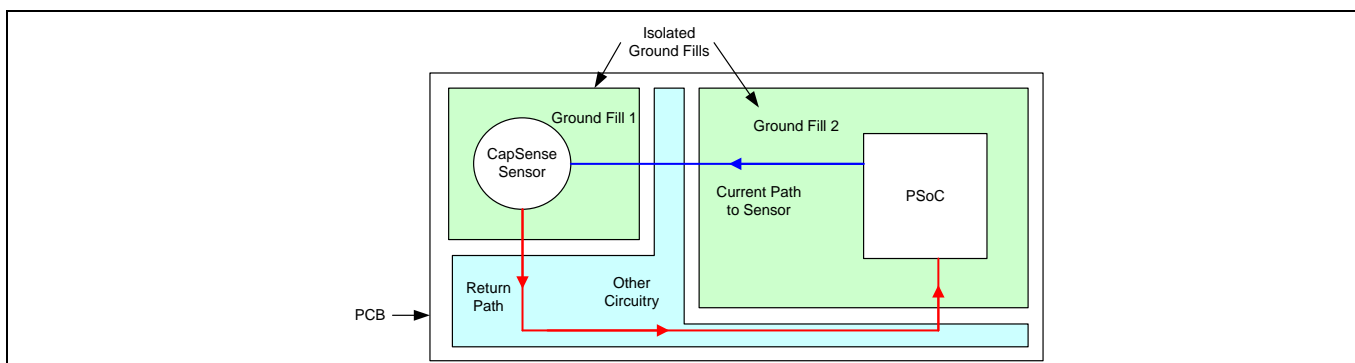


Figure 160 不適切な電流ループレイアウト

設計上の注意事項

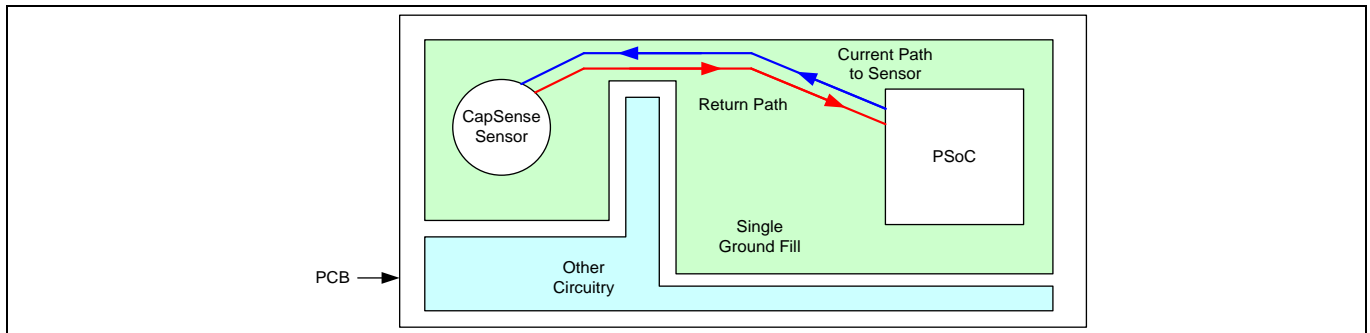


Figure 161 適切な電流ループレイアウト

RF ソースのロケーション

システムがスイッチング電源 (SMPS) やインバータなどの RF ノイズを発生させる回路を持つ場合、これらの回路を CAPSENSE™ インターフェースから遠く離れたところに配置する必要があります。また、そのような回路を保護して、RF 放射を低減する必要があります。Figure 162 は RF ノイズ源を CAPSENSE™ インターフェースから隔離する例を示します。

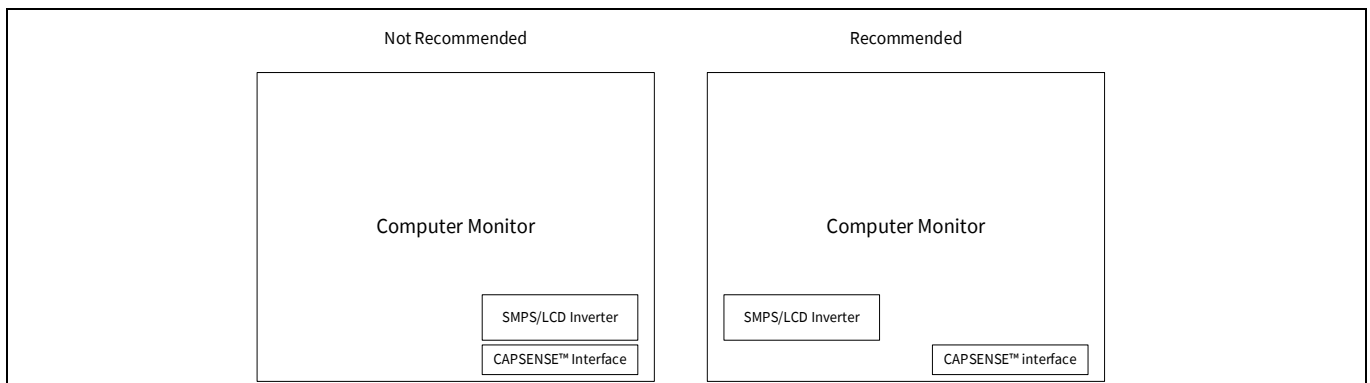


Figure 162 ノイズ源からの分離

ファームウェアの注意事項

CAPSENSE™ システムで放射性妨害波 (RE) は以下のパラメーターによって影響を受けます。

- デバイスの動作電圧
- デバイスの動作周波数
- センサー スイッチング周波数
- シールド信号
- センサー スキャン時間
- センスクロックソース非アクティブなセンサーの終端
- 各パラメーターの影響は以下の節で説明されます。

デバイスの動作電圧

放射はスイッチングが発生する電圧レベルに正比例します。如何なる出力ピンでのスイッチング信号の振幅がデバイスの動作電圧に直接左右されるため、動作電圧を低くすると、放射を大幅に削減するために役立ちます。

設計上の注意事項

PSoC™は低い電圧で動作できるため、放射を低減できます。Figure 163 および Figure 164 に動作電圧の放射性妨害波への影響を示します。IMO=24MHz であるため、24MHz で単一のスパイクがあり、他のスパイクはデバイスの異なるハードウェアおよびファームウェアの動作によって発生します。

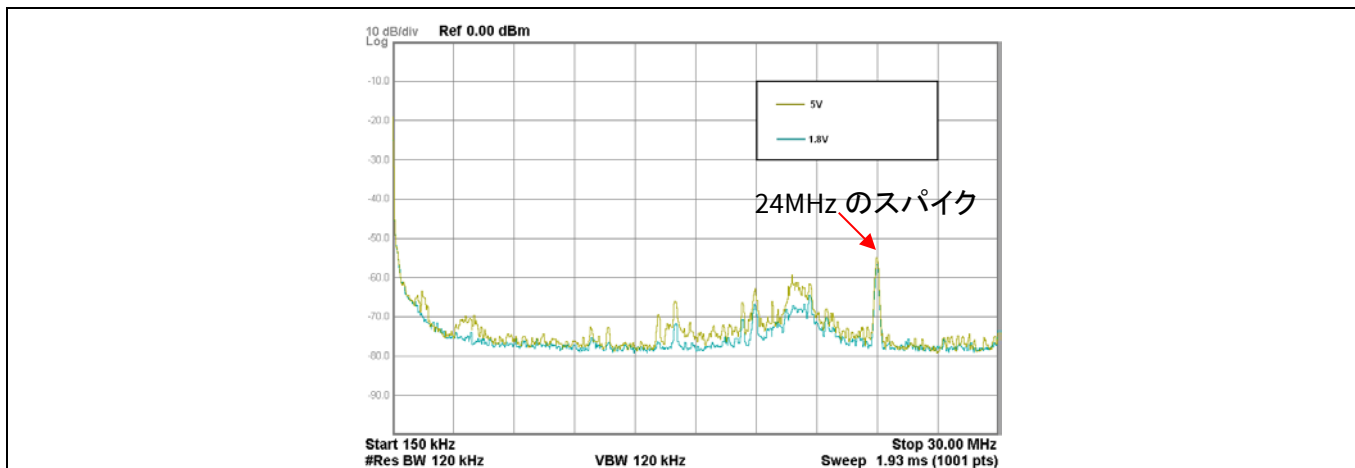


Figure 163 V_{DD} の放射性妨害波への影響 (150kHz~30MHz)

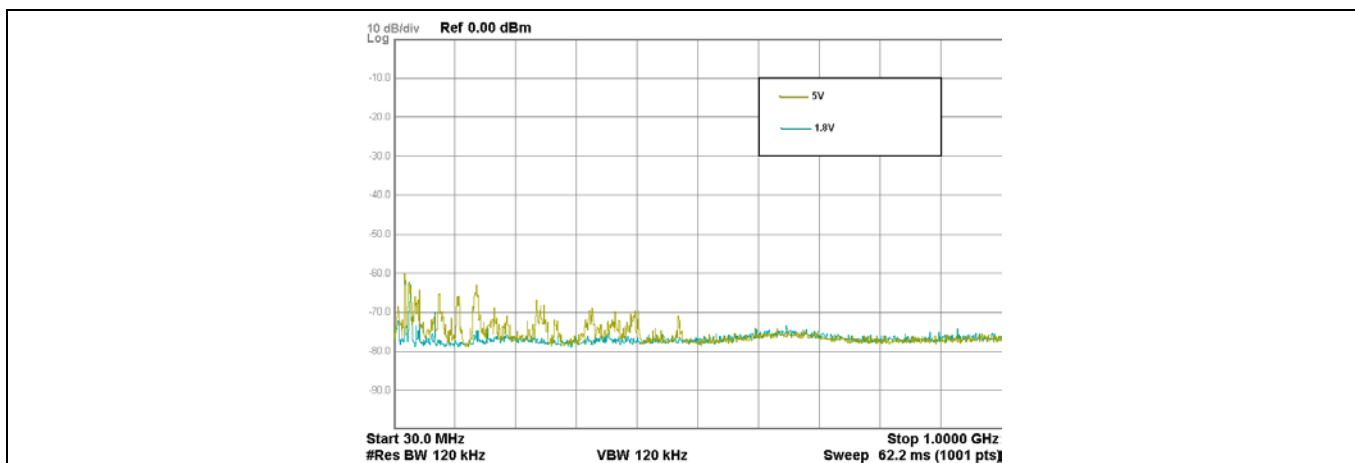


Figure 164 V_{DD} の放射性妨害波への影響 (30MHz~1GHz)

Note: 周波数軸は対数スケールです。

デバイスの動作周波数

システムクロック周波数 (IMO 周波数) を低くすると、放射性妨害波は低減されます。しかし、IMO 周波数は CPU クロックおよび他のシステムタイミングに与えるため、アプリケーションによっては IMO 周波数を下げられない場合があります。アプリケーションに応じて適切な IMO 周波数を選択します。

センサースイッチング周波数

センサースイッチング周波数を低くすると (センスクロックを参照してください)、放射性妨害波も低減されます。Figure 165 および Figure 166 を参照してください。IMO=24MHz であるため、24MHz で単一のスパイクがあり、他のスパイクはデバイスの異なるハードウェアおよびファームウェアの動作によって発生します。

設計上の注意事項

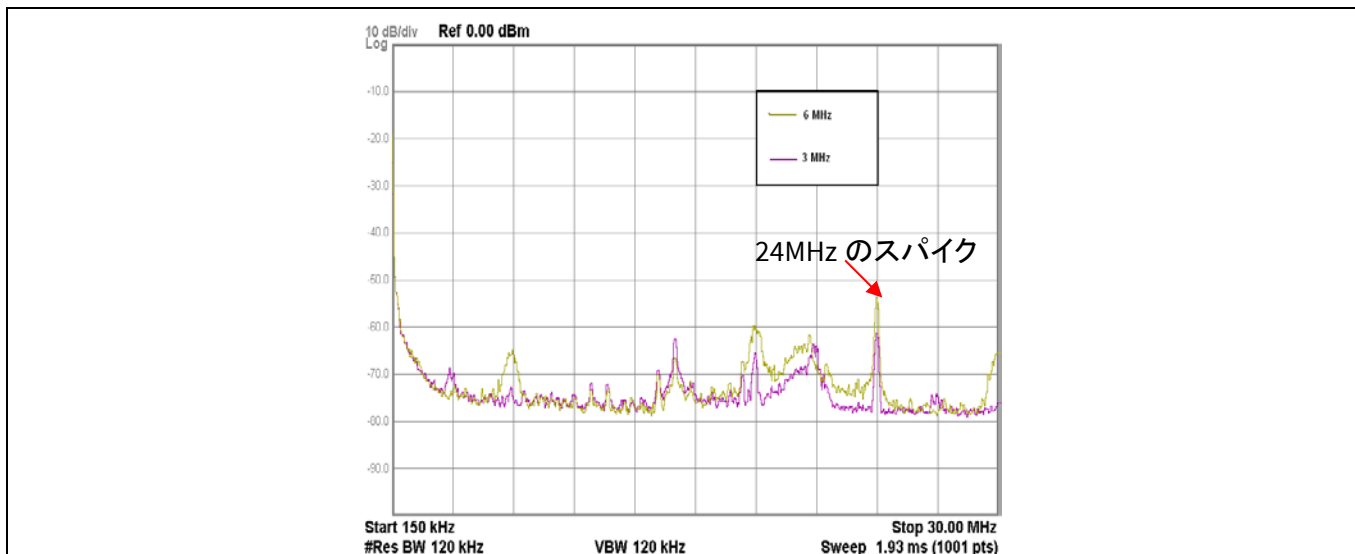


Figure 165 センサースイッチング周波数の放射性妨害波への影響 (150kHz~30MHz)

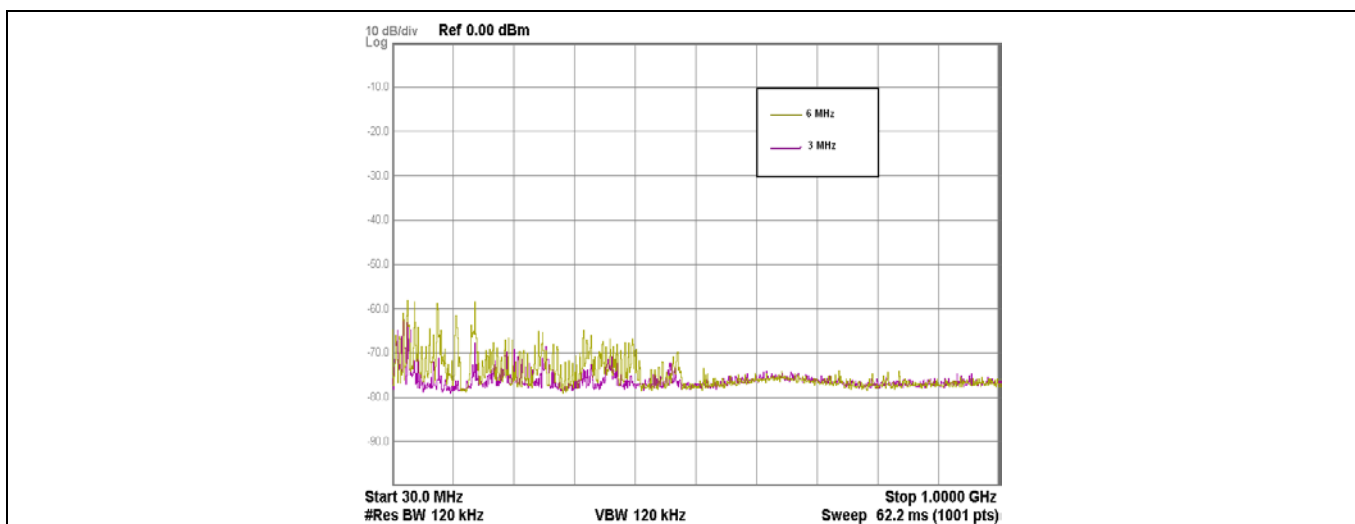


Figure 166 センサースイッチング周波数の放射性妨害波への影響 (30MHz~1GHz)

Note: 周波数軸は対数スケールです。

擬似ランダム センスクロック

PSoC™ 4 デバイスは PRS ベースのセンス クロック生成をサポートします。PRS は、固定クロックソースの代わりに使用され、固定周波数源によって生じる EMI 量を削減し、CAPSENSE™のピンに発生するノイズを減衰させ、他のソースやそれらの高調波から EMI 耐性を向上させます。

スペクトラム拡散センス クロック

PRS ベースのクロック生成に加えて、PSoC™ 4 S シリーズ, PSoC™ 4100S Plus, PSoC™ 4100PS, および PSoC™ 6 ファミリデバイスはスペクトラム拡散センス クロック生成という独特の機能をサポートし、これによりセンス クロック周波数が希望の範囲に広まります。この方法は特定の周波数範囲内でピーク数の削減とエミッションに有用です。センス クロックソースに **SSCn** を選択し、スペクトラム拡散クロックを有効にできます。周波数拡散の範囲はレジスタの長さ依存します。PSoC™ 4 S シリーズ, PSoC™

設計上の注意事項

4100S Plus, および PSoC™ 4100PS ファミリでのスペクトラム拡散センス クロック生成の詳細は、デバイスの該当 [テクニカル リファレンス マニュアル](#) で CAPSENSE™ 章のスペクトラム拡散センス クロックセクションを参照してください。

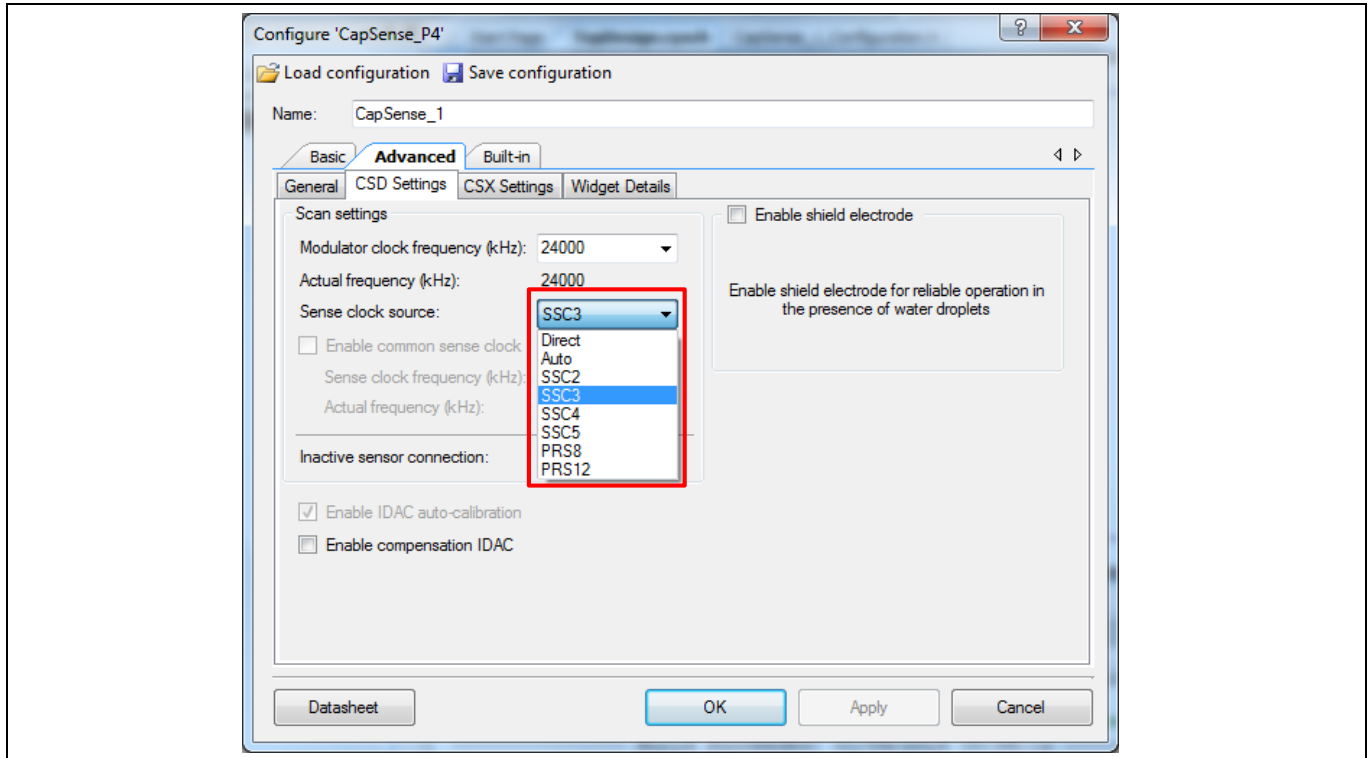


Figure 167 PSoC™ 4 S シリーズ, PSoC™ 4100S Plus, および PSoC™ 4100PS ファミリのセンス クロック ソース

シールド信号

ハッチパターンでシールド信号 ([被駆動シールド信号](#)および[シールド電極](#)を参照してください) を有効にすると、放射性妨害波は増加します。耐液性や近接センシングまたは高寄生容量の設計にのみ、被駆動シールド信号を有効にします。シールドを使用する必要がある場合、シールド電極領域がセンサーから 1cm の幅に制限されることを確認します ([Figure 148](#))。

センサーを囲むハッチパターンで被駆動シールド信号を有効にした際の放射性妨害波についての影響は [Figure 168](#) および [Figure 169](#) に示されます。これらの図で、被駆動シールド信号が無効にされたとき、ハッチパターンは接地されることに注意してください。IMO=24MHz であるため、24MHz で単一のスパイクがあり、他のスパイクはデバイスの異なるハードウェアおよびファームウェアの動作によって発生します。

設計上の注意事項

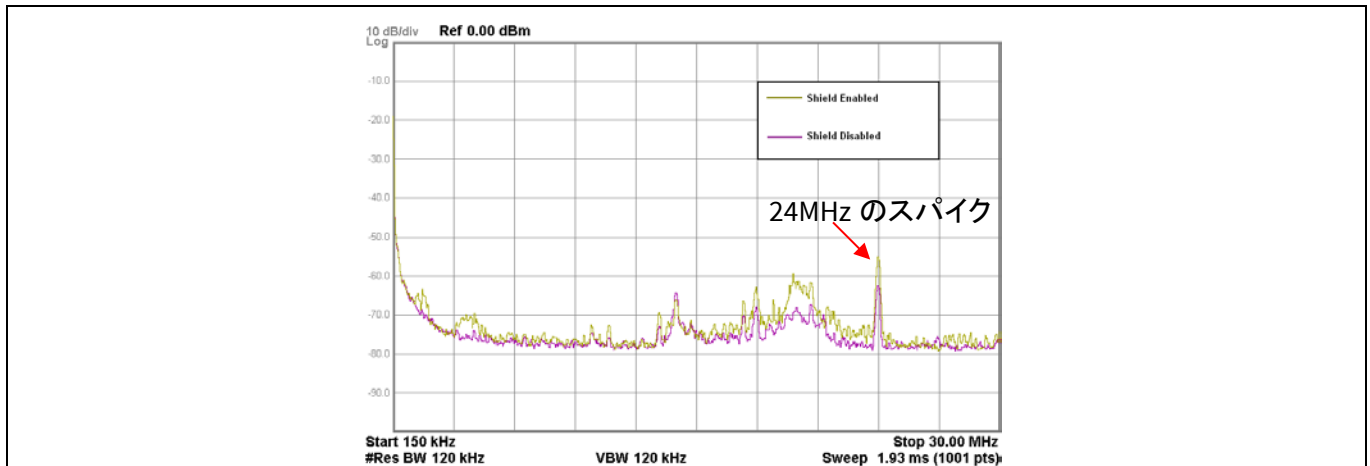


Figure 168 シールド電極の放射性妨害波への影響 (150kHz~30MHz)

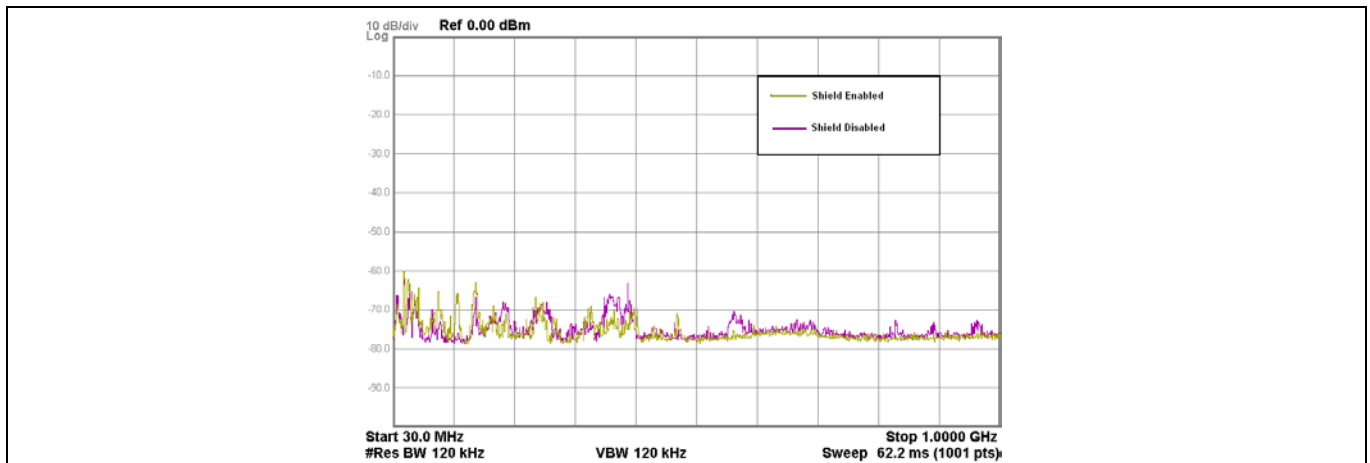


Figure 169 シールド電極の放射性妨害波への影響 (30MHz~1GHz)

Note: 周波数軸は対数スケールです。

センサー スキャン時間

センサー スキャン時間を短くすると、平均の放射性妨害波も減少されます。センサー スキャン時間はスキャン分解能および変調クロック分周器に依存します (式 9 を参照してください)。スキャン分解能または変調クロック分周器を高めると、スキャン時間も長くなります。Figure 170 および Figure 171 に、センサー スキャン時間の放射性妨害波への影響を示します。ここで、センサー スキャン時間がスキャン分解能を変更することによって変化することに注意してください。IMO=24MHz であるため、24MHz で単一のスパイクがあり、他のスパイクはデバイスの異なるハードウェアおよびファームウェアの動作によって発生します。

設計上の注意事項

Table 42 センサー スキャン時間

パラメーター	5 個のボタンの総スキャン時間	
	0.426ms	0.106ms
変調クロック分周器	2	2
スキャン分解能	10 ビット	8 ビット
個々のセンサー スキャン時間	0.085ms	0.021ms

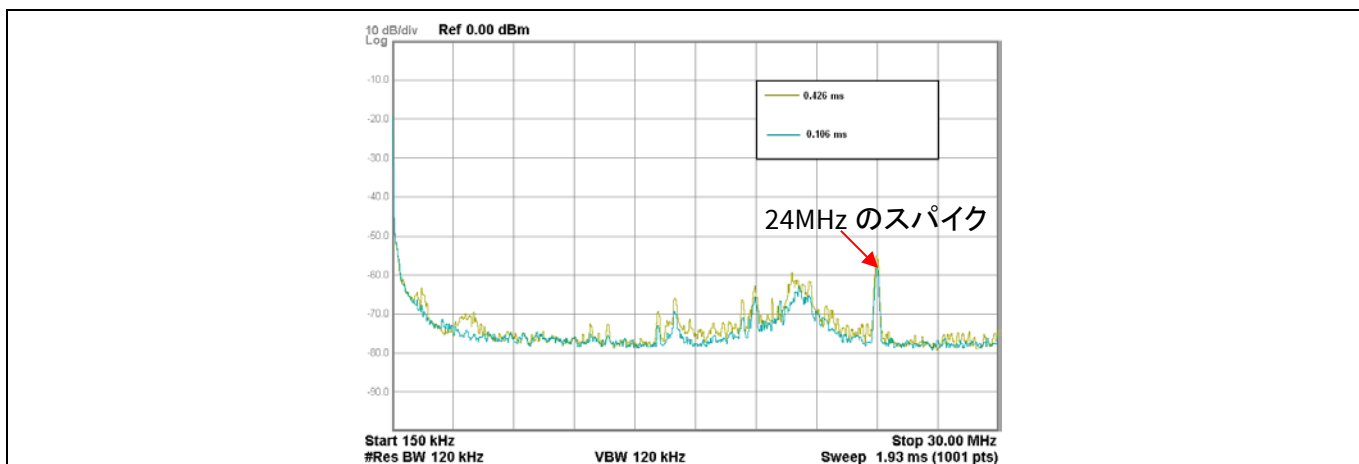


Figure 170 スキャン時間の放射性妨害波への影響 (150kHz~30MHz)

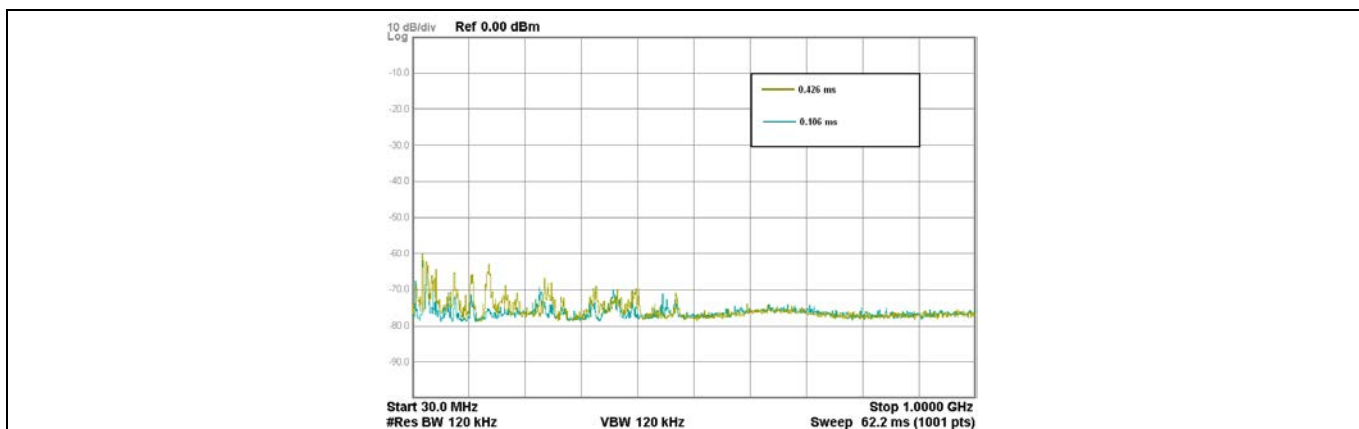


Figure 171 スキャン時間の放射性妨害波への影響 (30MHz~1GHz)

Note: 周波数軸は対数スケールです。

センスクロックソース

直接クロック駆動の代わりに PRS をセンスクロックソースとして使用する場合、放射スペクトラムが拡散され、そのため、平均の放射性妨害波が低減されます。Figure 172 および Figure 173 を参照してください。IMO=24MHz であるため、24MHz で単一のスパイクがあり、他のスパイクはデバイスの異なるハードウェアおよびファームウェアの動作によって発生します。

設計上の注意事項

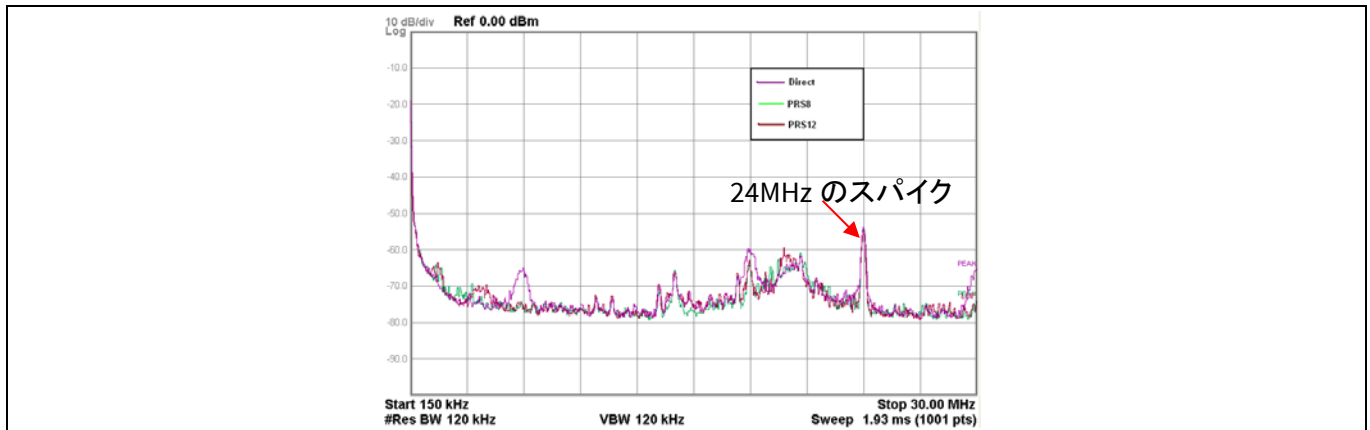


Figure 172 アナログ スイッチ駆動ソースの放射的妨害波への影響 (150kHz～30MHz)

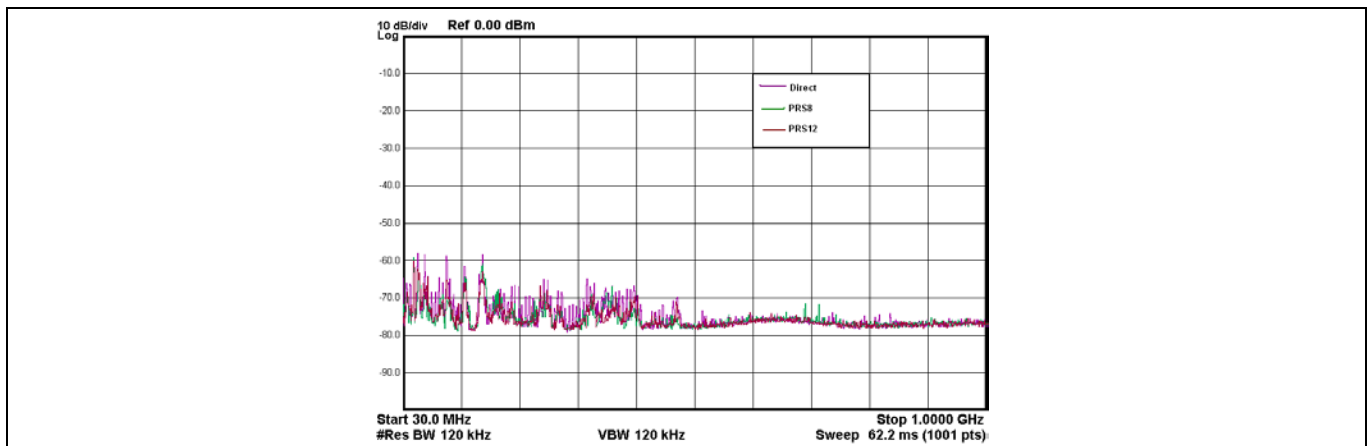


Figure 173 アナログ スイッチ駆動ソースの放射的妨害波への影響 (30MHz～1GHz)

Note: 周波数軸は対数スケールです。

非アクティブなセンサーの終端

放射を大幅に低下させるために、非アクティブなセンサーをシールドの代わりにグランドに接続します。Figure 174 および Figure 175 に、異なる非アクティブなセンサー終端の放射的妨害波への影響を示します。IMO = 24MHz であるため、24MHz で単一のスパイクがあり、他のスパイクはデバイスの異なるハードウェアおよびファームウェアの動作によって発生します。

設計上の注意事項

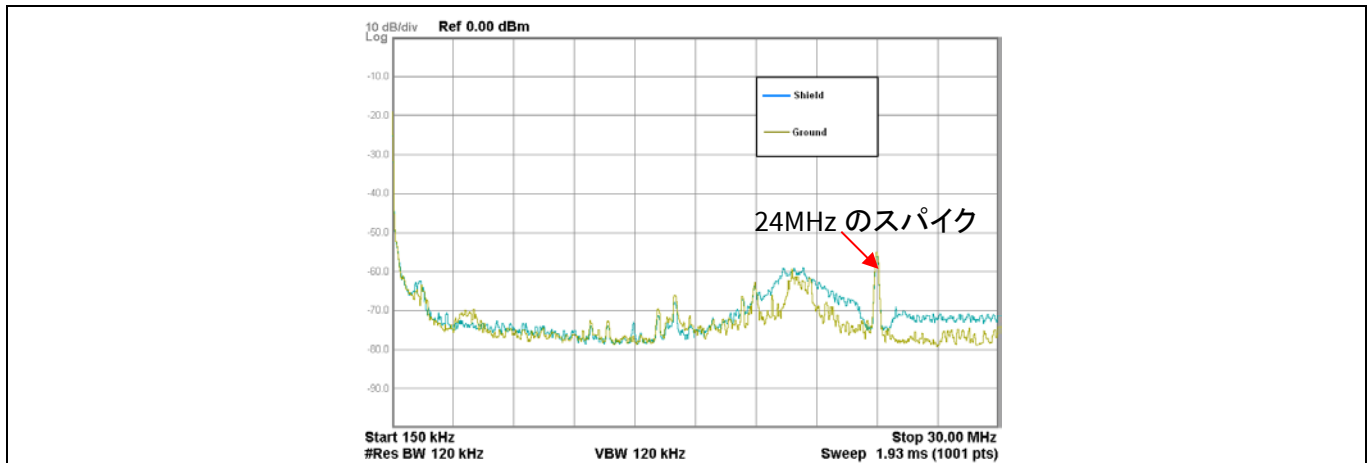


Figure 174 非アクティブなセンサーの放射的妨害波への影響 (150kHz~30MHz)

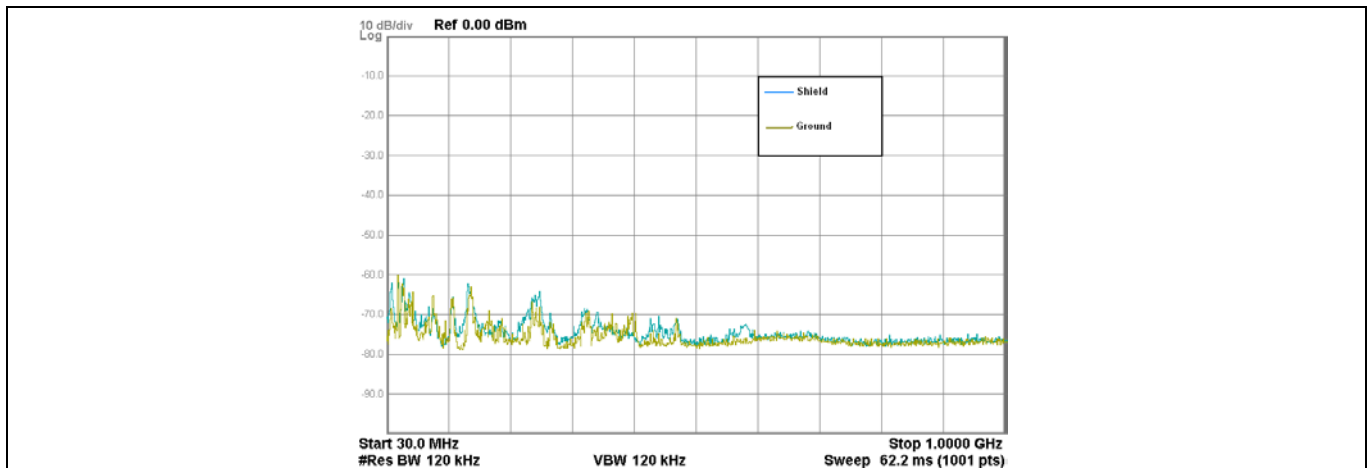


Figure 175 非アクティブなセンサーの放射的妨害波への影響 (30MHz~1GHz)

Note: 周波数軸は対数スケールです。

7.5.3.2.2 伝導性の RF ノイズ

電力および通信のラインを通じて CAPSENSE™ システムに入るノイズ電流は伝導性のノイズと呼ばれます。以下の技術を使用して伝導性の RF ノイズを抑えられます。

- 電源ピンのデカップリングコンデンサを使用して電源の伝導性ノイズを削減します。詳細は [7.4.11](#) および [デバイスデータシート](#) を参照してください。
- PCB の GND および VDD 面を提供して、電流ループを削減します。
- PSoC™ の PCB がケーブルで電源に接続されている場合は、ケーブル長を最短にして、シールドケーブルの使用を検討してください。

高周波ノイズを除去するために、電源や通信ラインにフェライトビーズを配置します。

設計上の注意事項

7.6 接地の影響

7.6.1 CSX 方式

指が CSX センサーに触れたときに CSX 方式で形成される等価静電容量を **Figure 176** に示します。**Figure 176** から、IDAC (I_{Rx}) から引き出される電流には、 I_{mt} と I_{sc} の 2 つのコンポーネントがあります。これらの 2 つのコンポーネントは、 C_{bodyDG}/C_{fs} の比率に依存します。raw カウントは IDAC から引き出される電流の量に依存するため、 C_{bodyDG}/C_{fs} の増減はセンサーの raw カウントに影響を与え、条件によっては動作に突然の変化を引き起こします。それをより理解するために、 $C_{bodyDG} \gg C_{fs}$ と $C_{bodyDG} \ll C_{fs}$ を引き起こす 2 つの極端な条件を考えてみましょう。

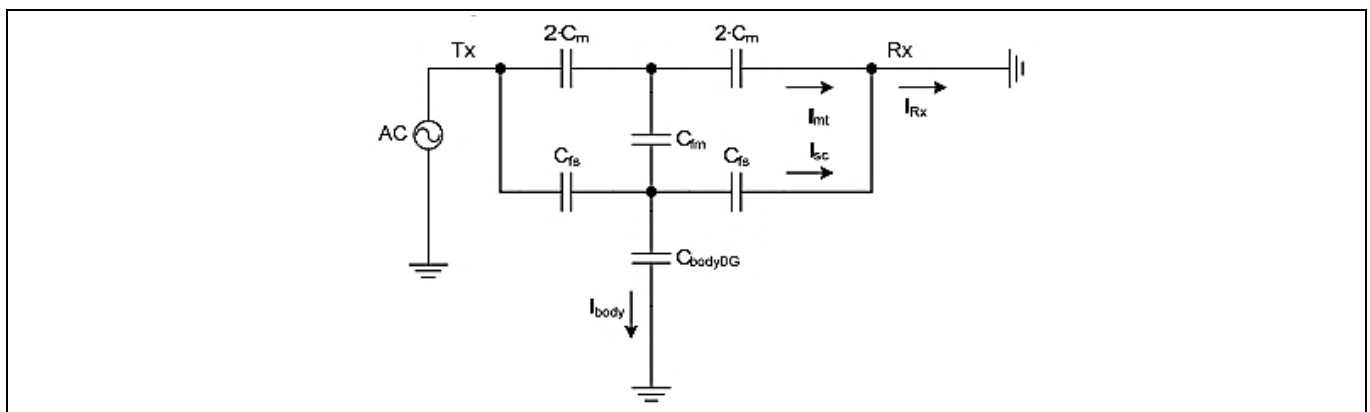


Figure 176 ボタンに指を置いたときの CSX センサーの等価回路

ここで、

C_M = Rx 電極と Tx 電極間の相互容量

C_{fs} = 指の表面と電極の間に形成される静電容量

C_{fm} = 指を置くことによって相互容量 C_M を減少させる仮想静電容量

C_{bodyDG} = デバイスのグラウンドに対するボディ容量

式 78. CSX 方式で IDAC から引き出される電流

$$I_{Rx} = I_{mt} + I_{sc}$$

I_{mt} は、Tx 電極と Rx 電極間の実効相互容量によるものです。

I_{sc} = センサーと指の間に形成された静電容量によって流れる寄生電流

7.6.1.1 CbodyDG >> Cfs

$C_{bodyDG} \gg C_{fs}$ であるため、 C_{bodyDG} を接地導体に置き換えられます。結果として得られる等価回路は、**Figure 177** に示すようになります。指で触れると、IDAC から引き出される電流は、Tx と Rx の間の実効相互容量に直接依存します。これは、良好なボード設計で観察される状態です。

設計上の注意事項

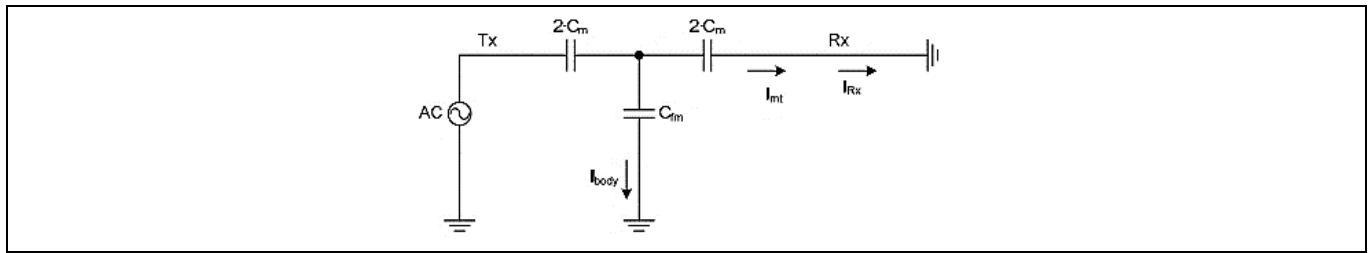


Figure 177 Cbody >> Cfs の場合の CSX センサーの等価回路

7.6.1.2 CbodyDG << Cfs

この状態 ($C_{bodyDG} \ll C_{fs}$) は、指が非常に薄いオーバーレイまたはオーバーレイなしで CSX ボタンに触れた場合、指が Rx および Tx 電極に直接接触した場合、または水滴が Rx および Tx 電極にのみ存在した場合に観察されます。 $C_{bodyDG} \ll C_{fs}$ なので、 C_{bodyDG} を削除できます。この場合の等価回路は Figure 178 のようになります。この状態では、指によって電極 C_{fs} に導入される静電容量は、デバイスの接地 C_{bodyDG} に対する指の静電容量と比較して非常に高くなります。

Figure 178 から、平衡ブリッジ回路を形成します。このため、 C_{fm} には電流が流れません。また、 C_{fs} の増加により、 I_{sc} が増加し、IDAC から追加の電流が引き出されます。これにより、raw カウントが減少するという予期しない動作が発生します。

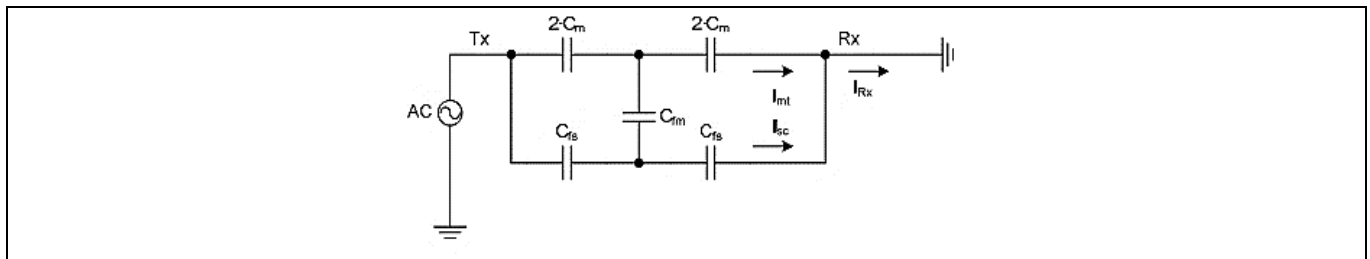


Figure 178 Cbody << Cfs の場合の CSX センサーの等価回路

CSX センサーの場合、設計では C_{bodyDG}/C_{fs} の比率を上げることに重点を置く必要があります。以下は C_{bodyDG}/C_{fs} の比率を増やすための例です。

1. C_{bodyDG}/C_{fs} 比は、オーバーレイの厚さ、センサーのサイズ、およびその他の多くの要因によって異なります。実験データによると、CSX センサーには 0.5mm 未満のオーバーレイの厚さを使用しないことを推奨します。オーバーレイの厚さを参照してください。
2. センサーがアースに接続されたハッチフィルで囲まれている場合、 $C_{bodyDG} \ll C_{fs}$ の可能性は低くなります。したがって、設計の適切な基盤を確保してください。ここで説明されている PCB レイアウトガイドラインのベストプラクティスに従ってください。
3. 設計では、直接の相互作用を避けるために、CSX タッチセンシングシステムの Rx および Tx 電極、外部コンデンサ、および抵抗器のトレースラインを導電面または指のタッチから分離することを推奨します。この推奨事項に従わない場合、 $C_{bodyDG} \ll C_{fs}$ が発生する可能性があります。

7.6.2 CSD 方式

指が CSD センサーに触れたときに CSD 法で形成される等価静電容量を Figure 179 に示します。これは、IDAC から引き出される電流が、指で触れることによって導入される静電容量に直接依存することを示します。 I_{CP} は固定コンポーネントであり、 I_{CF} は C_F, C_{BG}, C_{GE} に依存します。式 10 から、raw カウント

設計上の注意事項

は IDAC から引き出される電流の量に依存します。それをより理解するために、AC/DC 主電源アプリケーションとバッテリー電源アプリケーションの 2 つのシナリオを考えてみましょう。

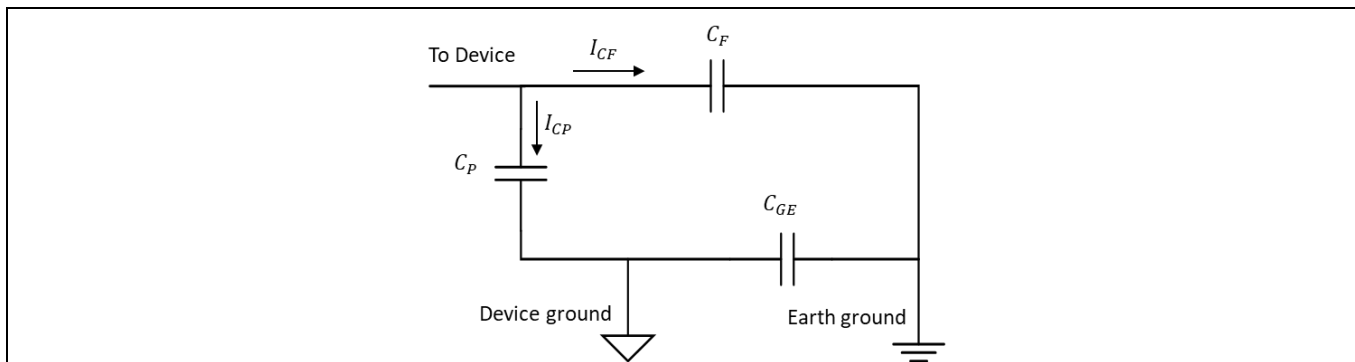


Figure 179 CSD センサーの等価回路

式 79. CSD 方式で IDAC から引き出される電流

$$I = I_{CP} + I_{CF}$$

7.6.2.1 AC/DC 駆動のアプリケーション

主電源を使用する AC/DC 電源のアプリケーションでは、デバイスのグラウンドはアースのグラウンドに強く結合されます。したがって、C_{GE} を導体に置き換えができ、C_{BG} は通常 100pF から 200pF です。C_{BG} は C_F に比べて大きいため、その影響は無視できます。最後に、結果の等価回路を Figure 180 に示します。総静電容量の増加により、IDAC からより多くの電流が引き出され、指で触れた場合の raw カウントの変化が大きくなります。したがって、この状態では、感度が高くなります。つまり、指で触れたときの信号が高くなります。

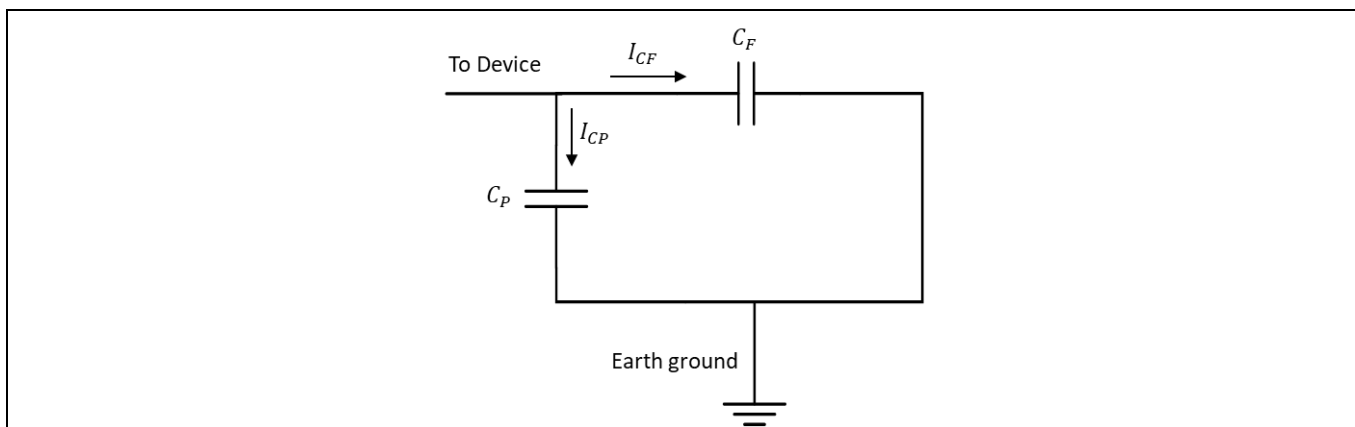


Figure 180 主電源アプリケーション用の CSX センサーの等価回路

7.6.2.2 バッテリー駆動のアプリケーション

バッテリー駆動のポータブルアプリケーションでは、デバイスのグラウンドとアースのグラウンドが軽く結合されているため、C_{GE} は小さくなります。結果として得られる等価回路を Figure 181 に示します。したがって、この状態では、感度が低くなります。つまり、デバイスで見られる静電容量の減少により、指で触れると信号が低くなります。

設計上の注意事項

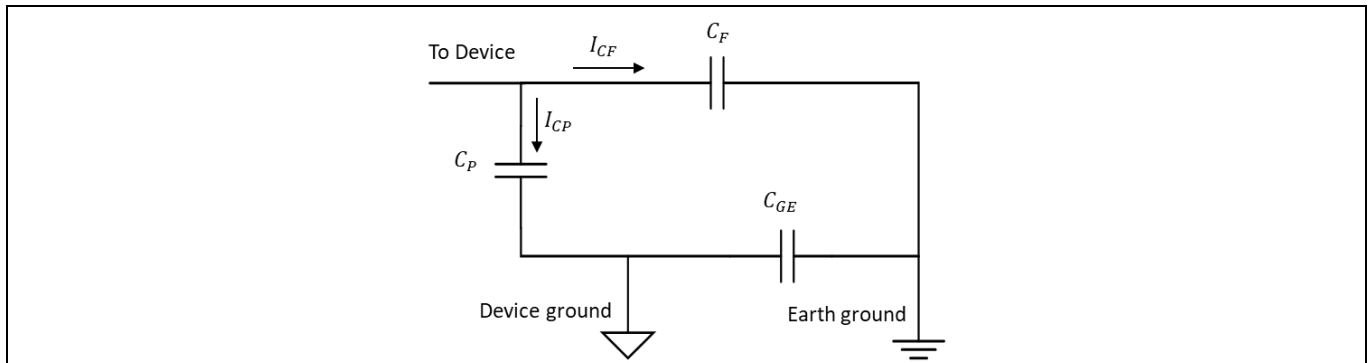


Figure 181 バッテリ駆動アプリケーション用の CSD の等価回路

以下は、バッテリーを使用するポータブルアプリケーションでの CSD システム設計の推奨事項です。

1. システムに大きなグランドプレーンを追加します。グランドプレーンは、センサーの寄生容量を増加させないように、検出要素から離しておく必要があります。この章で説明されている [PCB レイアウトガイドライン](#) のベストプラクティスに従ってください。
2. 被駆動シールドを使用して、ポータブルデバイスの感度を向上させます。詳細については、[シールド電極のレイアウトガイドライン](#) を参照してください。
3. オーバーレイ材料の厚さを薄くするか、誘電率の高いオーバーレイを使用して感度を向上させます。
4. バッテリ電源から電力を供給して CAPSENSE™ システムを調整します。

CAPSENSE™ Plus

8 CAPSENSE™ Plus

PSoC™ 4 は CAPSENSE™ と一緒に多くの追加機能を実行できます。Figure 182 に示すように、このデバイスが提供しているさまざまな機能により、さまざまなシステム機能を 1 つのチップに組み込みます。このようなアプリケーションは CAPSENSE™ Plus アプリケーションとして知られます。

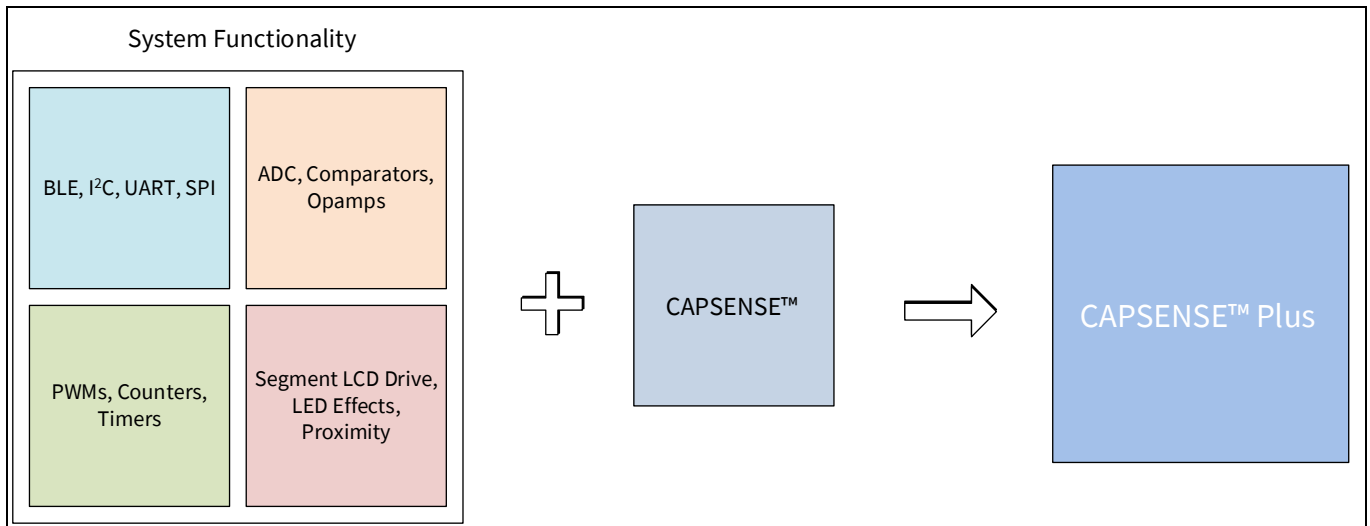


Figure 182 CAPSENSE™ Plus

PSoC™ 4 デバイスで使用可能な追加機能は以下のとおりです。

- 通信: Bluetooth® LE, I2C, UART, SPI, CAN, および LIN
- アナログ機能: ADC, コンパレータ, およびオペアンプ
- デジタル機能: PWM, カウンター, タイマー, および UDB
- セグメント LCD 駆動
- ブートローダー
- 各種消費電力モード: アクティブ, スリープ, ディープスリープ, ハイバネート, およびストップ

上記の追加機能を使用する場合は、必要に応じてシンクモードで設定することを推奨します。

PSoC™ 4 の詳細については [AN79953 - Getting started with PSoC™ 4](#)、または [AN91267 - Getting started with PSoC™ 4 Bluetooth® LE](#) を参照してください。

PSoC™ 4 および固有の PSoC™ Creator IDE の柔軟性により、設計をすぐに変更でき、すなわち市場投入までの時間を短縮できます。他のシステム機能を統合すると、システム全体のコストを大幅に削減できます。Table 43 に CAPSENSE™ Plus の使用によりコストを削減できるサンプルアプリケーションの一覧を示します。

CAPSENSE™ Plus

Table 43 CAPSENSE™ Plus の例

応用	CAPSENSE™	オペ アンプ	ADC	コンパ レータ	PWM, カウ ンター, タ イマー, UDB	通信 (Bluetooth® LE, I2C, SPI, UART)	LCD 駆動	GPIO
心拍数モ ニター (リスト バンド)	ユーザー インターフ ェース: ボタン, リニ ア スライダー	TIA, バッ ファ	心拍数 の測 定、電 池電圧 の測定		LED 駆動	Bluetooth® LE	セグ メン ト LCD	LED 表示
LED バル ブ	ユーザー インターフ ェース: ボタン, ラジ アル スライダー	アン プ	LED 電 流測定	短絡回 路の保 護	LED 色の制 御 (PrISM ¹⁸)	Bluetooth® LE		LED 表示
洗濯機	ユーザー インターフ ェース: ボタン, ラジ アル スライダー		温度セ ンサー	水位監 視計	ブザー、 FOC ¹⁹ モ ーター制御	I ² C LCD ディス プレイ、UART ネットワーク インターフェ ース	セグ メン ト LCD	LED 表示
温水器	ユーザー インターフ ェース: ボタン, リニ ア スライダー		温度セ ンサー、水 流セン サー	水位観 測計	ブザー	I ² C LCD ディス プレイ、UART ネットワーク インターフェ ース	セグ メン ト LCD	LED 表示
IR リモコ ンコン トローラ ー	ユーザー インターフ ェース: ボタン, リニ ア および ラジアルの スライダー, タッチ パッド				Manchester エンコー ダ			LED 表示
IH 調理 器	ユーザー インターフ ェース: ボタン, リニ ア スライダー		温度セ ンサー				セグ メン ト LCD	LED 表示
モーター 制御シス テム	ユーザー インターフ ェース: ボタン, リニ ア スライダー				BLDC ²⁰ お よび FOC モーター 制御			LED 表示
ゲーム/ シミュレ ーション コントロ ーラー	ユーザー インターフ ェース: ボタン, タッ チパッド		アナロ グ読出 しジョ イスティック			I ² C/SPI/UART 通信インター フェース	セグ メン ト LCD	LED 表示
感熱式プ リンタ	ユーザー インターフ ェース: ボタン		加熱保 護、紙セ ンサー		ステッピ ングモ ーター制御	SPI 通信イン ターフェース		LED 表示

¹⁸ PrISM = Precision Illumination Signal Modulation (高精度照度信号変調)

¹⁹ FOC = Field Oriented Control (フィールド指向制御)

²⁰ BLDC = Brushless DC Motor (ブラシレス DC モーター)

CAPSENSE™ Plus

Figure 183 に、IH 調理器や電子レンジなどの CAPSENSE™ Plus アプリケーションの基本ブロック図を示します。

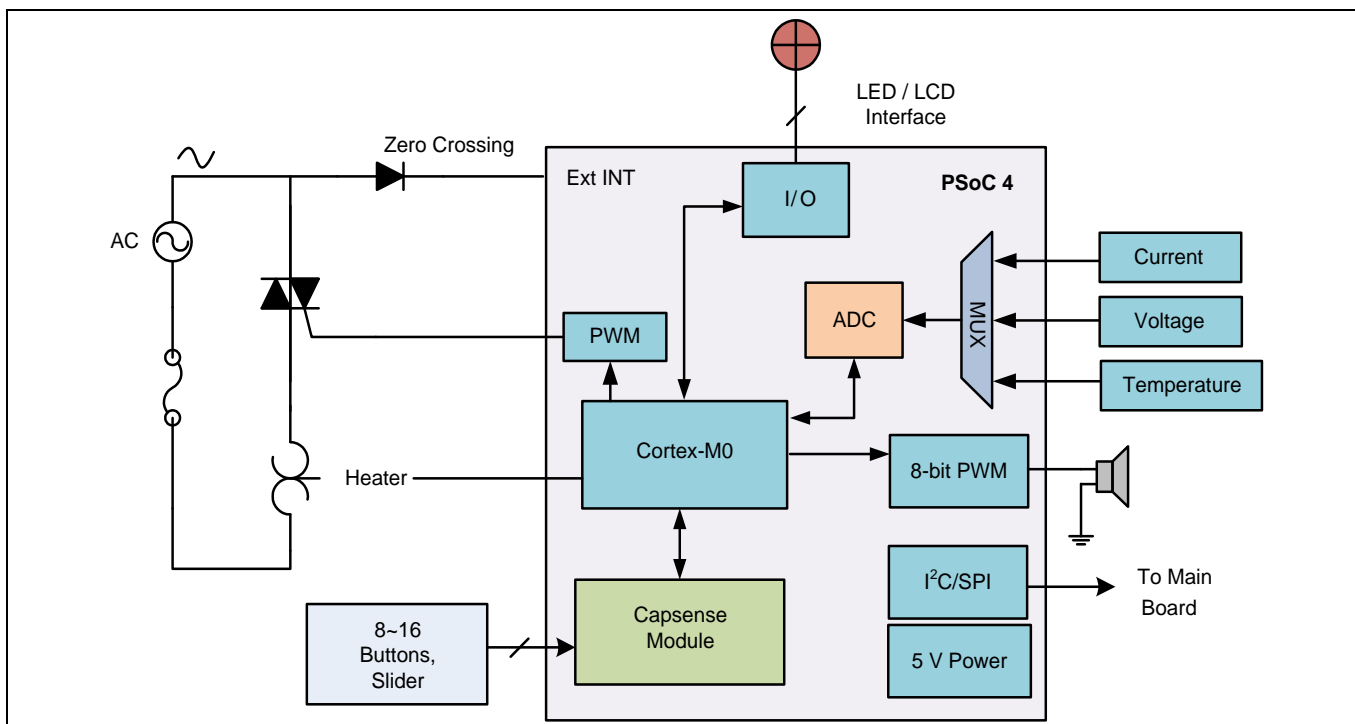


Figure 183 PSoC™ 4 の CAPSENSE™ Plus システム

本アプリケーションでは、PSoC™ 4 の 12 ビット 1Msps SAR ADC は過電流、過電圧および高温状態を検出します。PWM 出力は、ステータスとアラーム音のスピーカーを駆動します。別の PWM は、システムの加熱素子を制御します。CAPSENSE™ ボタンおよびスライダはユーザーインターフェースを構成します。PSoC™ 4 は視覚化出力のためにセグメント LCD も駆動できます。PSoC™ 4 は、システムのメインボードに接続できるシリアルコミュニケーションブロックを持ちます。

Figure 184 に、Bluetooth® LE 接続搭載の PSoC™ 6 MCU に基づくフィットネストラッカーのアプリケーションレベルのブロック図を示します。このデバイスはワンチップソリューションを提供し、アクティビティ モニタリング、環境モニタリング、ユーザーインターフェース用 CAPSENSE™、Bluetooth® LE 接続などの機能を含みます。PSoC™ 6 MCU の詳細な情報は [AN210781 - Getting started with PSoC™ 6 MCU with Bluetooth® LE connectivity](#) を参照してください。

CAPSENSE™ Plus

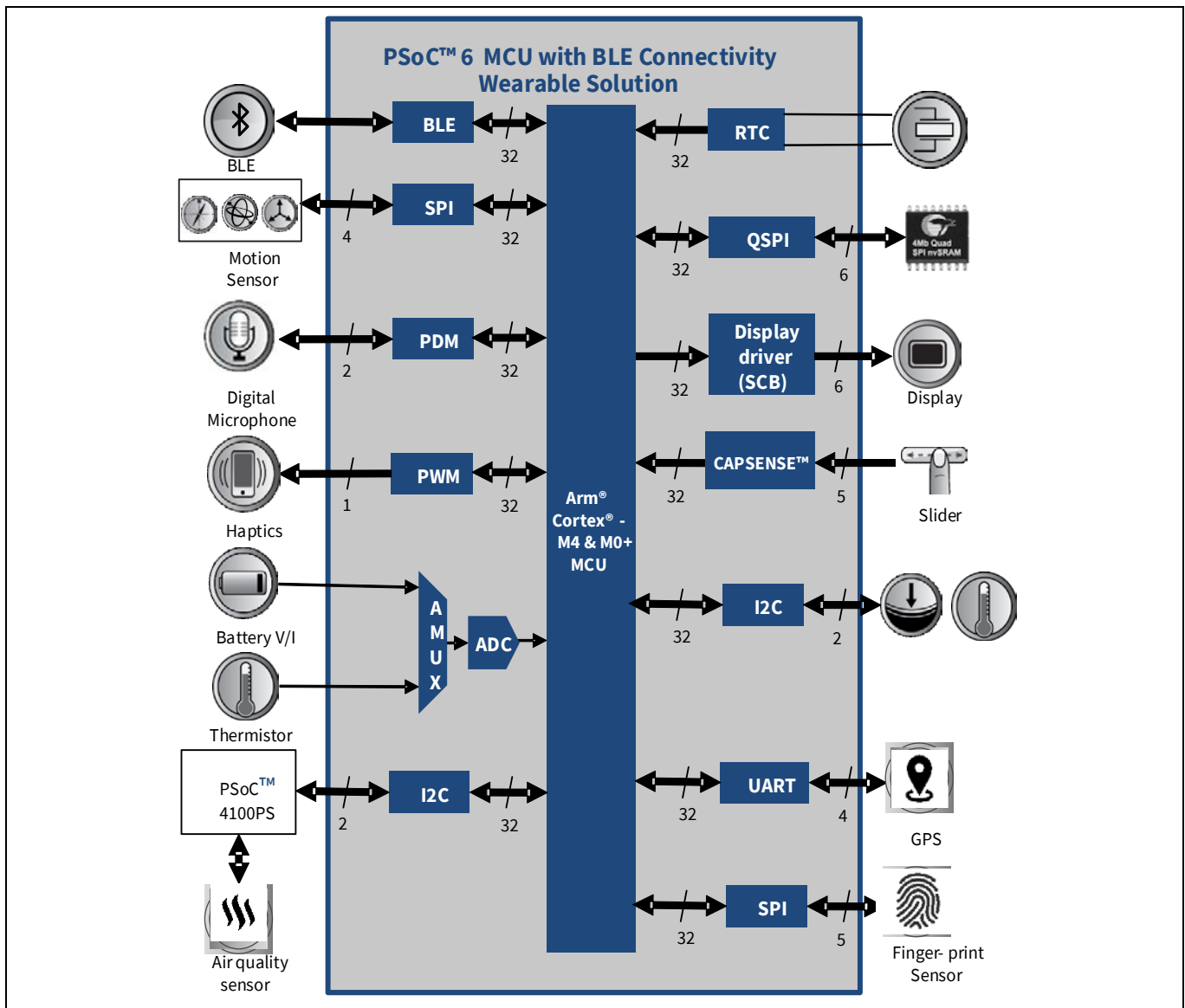


Figure 184 Bluetooth® LE 接続搭載の PSoC™ 6 MCU に基づくフィットネストラッカーのアプリケーションのブロック図

上記の 2 つ例のような CAPSENSE™ Plus システムにより、基板サイズ, BOM コスト, および電力消費を削減できます。

リソース

9 リソース

9.1 ウェブサイト

Bluetooth® LE 接続搭載の PSoC™ 4 および PSoC™ 6 MCU の詳細は、[Getting started with PSoC™ 4](#), [Getting started with PSoC™ 4 Bluetooth® LE](#), [Getting started with PSoC™ 6 MCU](#), および [Getting started with PSoC™ 6 MCU with Bluetooth® LE connectivity](#) ウェブサイトをご覧ください。

9.2 デバイス データシート

- [PSoC™ 4 データシート](#)
- [PSoC™ 4 Bluetooth® LE データシート](#)
- [PSoC™ 6 MCU デバイス](#)

9.3 コンポーネント データシート/ミドルウェア文書

- [PSoC™ 4 capacitive sensing](#)
- [PSoC™ 6 capacitive sensing](#)
- [CAPSENSE™ middleware library](#)
- [ModusToolbox™ CAPSENSE™ configurator guide](#)

9.4 テクニカル リファレンス マニュアル

[PSoC™ 4 Technical reference manual \(TRM\)](#) および [PSoC™ 6 Technical reference manual \(TRM\)](#) では、上位レベル アーキテクチャ図、レジスタのサマリー、およびタイミング図などの PSoC™ 4 と PSoC™ 6 アーキテクチャに関する情報を素早く簡単に取得できます。

9.5 開発キット

Table 6、PSoC™ 4 および PSoC™ 6 CAPSENSE™ に対応するインフィニオン® 開発キットの一覧を示します。

9.6 PSoC™ Creator

PSoC™ Creator は最先端で使いやすい統合開発環境です。[PSoC™ Creator ホームページ](#)を参照してください。

9.7 ModusToolbox™

PSoC™ 4 および PSoC™ 6 ベースの CAPSENSE™ アプリケーションを開発するために ModusToolbox™ ソフトウェアスイートが使用されます。[こちら](#)から ModusToolbox™ ソフトウェアをダウンロードできます。関連資料は以下のとおりです。

- [ModusToolbox™ release notes](#)
- [ModusToolbox™ install guide](#)
- [ModusToolbox™ user guide](#)
- [ModusToolbox™ quick start guide](#)
- [ModusToolbox™ CAPSENSE™ configurator](#)
- [ModusToolbox™ CAPSENSE™ tuner](#)
- [ModusToolbox™ device configurator](#)

リソース

- [ModusToolbox™ SmartIO configurator](#)
- [PSoC™ Creator to ModusToolbox™](#)
- [ModusToolbox™ command line](#)

9.8 アプリケーションノート

設計を速やかに立ち上げて実行するために大量のアプリケーションノートが利用可能です。[PSoC™ 4 application notes](#), [PSoC™ 4 Bluetooth® LE application notes](#), [CAPSENSE™ application notes and design guides](#) を参照してください。

CAPSENSE™ に特有のアプリケーションノートの一覧は以下のとおりです。

PSoC™ 3 および PSoC™ 5LP デバイス用のデザインガイド

- [PSoC™ 3 and PSoC™ 5LP CAPSENSE™ design guide](#)

CAPSENSE™ Express ファミリ用のデザインガイド

- [CY8CMBR3XXX CAPSENSE™ design guide](#)
- [CY8CMBR2110 CAPSENSE™ design guide](#)
- [CY8CMBR2016 CAPSENSE™ design guide](#)
- [CY8CMBR2010 CAPSENSE™ design guide](#)
- [CY8CMBR2044 CAPSENSE™ design guide](#)
- [CAPSENSE™ Express™: CY8C201XX application notes](#)

PSoC™ 1 デバイス用のデザインガイド

- [CY8C20XX7/S design guide](#)
- [CY8C20XX6A/H CAPSENSE™ design guide](#)
- [CY8C21X34/B CAPSENSE™ design guide](#)
- [CY8C20X34 CAPSENSE™ design guide](#)

入門アプリケーションノート

- [AN79953 - Getting started with PSoC™ 4](#)
- [AN210781 - Getting started with PSoC™ 6 MCU with Bluetooth® LE connectivity](#)
- [AN221774 - Getting started with PSoC™ 6 MCU](#)

9.9 設計サポート

- **知識ベース記事:** 製品ファミリ別の技術情報記事を読覧したり、CAPSENSE™ についてのさまざまなトピックを検索できます。
- **ホワイトペーパー:** 高度な静電容量タッチインターフェースに関するトピックについて学べます。
- **サイプレス開発コミュニティ:** 技術コミュニティに参加し、情報交換できます。
- **ビデオライブラリ:** チュートリアルビデオで素早く学習できます。
- **品質および信頼性:** 当社は顧客満足を第一に考えます。当社の品質ウェブサイトでは、信頼性および製品の品質レポートをご覧になれます。
- **テクニカルサポート:** サポートケースを作成してレビュー対象の設計を提出します。[テクニカルサポート](#) に繋がるためには、ウェブサイト登録し、ログインしてください。レイアウトの階層情報付き回路図とガーバーファイルについては PDF 印刷の使用を推奨します。

用語集

10 用語集

AMUXBUS

I/O ピンを複数の内部アナログ信号に接続する PSoC™ 内にあるアナログ マルチプレクサ バスです。

ベースライン

センサーに人の指で触らないときの raw カウントの傾向を推定しファームウェア アルゴリズムから得られる値です。ベースラインは raw カウントの突然の変化に敏感性が低く、差分カウントを計算するためのリファレンス点を提供します。

ボタンまたはボタン ウィジェット

センターに対応しており、センサーのアクティブ状態または非アクティブ状態 (すなわち、2 つだけの状態) を報告するウィジェットです。例えば、センサー上の指の「タッチあり」または「タッチなし」の状態を検出できます。

差分カウント

raw カウントとベースラインの差です。差分値が負であるかまたはノイズしきい値未満である場合、差分カウントは常に 0 に設定されます。

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 (プリント回路基板 (PCB) 上の銅ボタンなど) です。

CAPSENSE™

インフィニオンのタッチセンシング ユーザー インターフェイス ソリューションです。業界 2 位に対して、4 倍の販売実績がある業界 No.1 ソリューションです。

CAPSENSE™ メカニカル ボタン リプレースメント (MBR)

メカニカル ボタンを静電容量ボタンにアップグレードするサイプレスの構成可能なソリューションであり、センサー パラメーターの設定に必要な設計工数を最小限に抑え、ファームウェアの開発も不要とします。これらのデバイスは CY8CMBR3XXX および CY8CMBR2XXX のファミリーを含みます。

重心/重心位置

スライダー分解能の指定した範囲内のスライダー上の指の位置を示す数です。この数は CAPSENSE™ 重心計算アルゴリズムにより算出されます。

補正 IDAC

過剰なセンサー C_p 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です。この IDAC は変調 IDAC と違って、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 (CSD) は、静電容量センシングのアプリケーション用に自己容量を測定する特許取得済みの方法です。

CSD モードでは、センシングシステムは電極の自己容量を測定し、指の有無を識別するために自己容量の変化が検出されます。

用語集

デバウンス

有効なタッチとなるためにタッチがある必要な連続スキャン サンプル数を定義するパラメーターです。このパラメーターは怪しいタッチ信号を除去するために役立ちます。

指のタッチは、差分カウントがスキャン サンプルの連続デバウンス数 (指しきい値+ヒステリシス) を超える場合にのみ報告されます。

被駆動シールド

シールド電極がセンサー スイッチング信号と同じ位相および振幅を持つ信号によって駆動され、耐液性を有効にするために CSD によって使用される技術です。

電極

PCB、ITO または FPCB 上のパッドや層などの導電材料です。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。

指しきい値

センサーの状態を確定するためにヒステリシスと一緒に使用されるパラメーターです。センサーの状態は、差分カウントが (指しきい値+ヒステリシス) を上回る場合にオンとして報告され、差分カウントが (指しきい値 - ヒステリシス) を下回る場合にオフとして報告されます。

連動センサー

複数のセンサーを連動させ、単一センサーとしてスキャンする方法です。近接センシング用のセンサーの領域を増やし、消費電力を減少させ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ときに消費電力を削減するために、センサーは個別にスキャンされず、すべてを連動して単一のセンサーとしてスキャンされ、時間を短縮させます。ユーザーがセンサーのいずれかをタッチすると、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™ はファームウェアによるセンサー連動をサポートします。すなわち、複数のセンサーがスキャンのために AMUXBUS に同時に接続できます。

ジェスチャー

ジェスチャーはスワイプやピンチズームなどのユーザーの行動です。CAPSENSE™ は事前に定義されたタッチ パターンに基づいて異なるジェスチャーを識別するジェスチャー検出機能を備えています。CAPSENSE™ コンポーネントでは、ジェスチャー機能はタッチパッド ウィジェットのみにによってサポートされます。

ガード センサー

ボタンセンサーと同様に PCB 上のすべてのセンサーを取り囲み、液体流を検出するために使用される銅配線です。ガードセンサーがトリガーされると、ファームウェアは誤ったタッチを防ぐために、すべての他のセンサーのスキャンを無効にできます。

ハッチ フィル/ハッチ グランド/ハッチド グランド

静電容量センシングの PCB を設計する際、良好なノイズ耐性のために接地した銅面をセンサーの周りに配置する必要があります。しかし、ベタ グランドはセンサーの期待されない寄生静電容量を増加させます。そのため、グランドは特別なハッチ パターンで充填する必要があります。ハッチ パターンはメッシュのように密接に配置され、交差されるラインがあり、充填率がラインの幅および 2 本のライン間の

用語集

間隔によって決まります。耐液性の場合、シールド電極として呼ばれるこのハッチ フィルはグラウンドの代わりにシールド信号で駆動されます。

ヒステリシス

システム ノイズに起因してセンサー状態の出力がランダムにトグルすることを回避し、センサーの状態を決定するために指しきい値と一緒に使用されるパラメーターです。指しきい値を参照してください。

IDAC (電流出力デジタル-アナログ変換器)

CAPSENSE™ および ADC 動作の PSoC™ 内のプログラマブルな定電流源です。

耐液性

水滴、液体流や霧が存在する環境でも確実に動作する静電容量センシング システムの能力です。

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を含むウィジェットです。

低ベースラインリセット

raw カウントが負のノイズしきい値を異常に下回るスキャン サンプルの最大数を表すパラメーターです。低ベースラインリセットの値を超える場合、ベースラインは現時点の raw カウントにリセットされます。

手動チューニング

CAPSENSE™ パラメーターの手動設定 (または手動チューニング) プロセスです。

マトリックス ボタン

マトリックス状で配置された 2 つ以上のセンサーから成り、垂直方向および水平方向に配置されるセンサーの交差部に人の指 (タッチ) の有無を検出するために使用されるウィジェットです。

M を水平軸上のセンサーの数と、N を垂直軸上のセンサーの数とすれば、マトリックス ボタン ウィジェットは (M+N) 本のポート ピンだけを使用して合計で M x N 個の交差部を監視できます。

CSD センシング方式 (自己容量) を使用する場合、このウィジェットは同時に 1 点のみの交差位置で有効なタッチを検出できます。

変調コンデンサ (C_{MOD})

自己容量センシング モードでの CSD ブロックの動作のために必要な外部コンデンサです。

変調クロック

センサーのスキャン間に CSD ブロックから変調器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です。このクロックが raw カウント 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 $[(2^N - 1) / \text{変調クロック周波数}]$ (N はスキャンの分解能) により計算されます。

変調 IDAC

変調 IDAC はプログラマブルな定電流源であり、この出力は V_{REF} の AMUXBUS 電圧を維持するために、CSD ブロックのシグマデルタ変調器の出力によって制御 (オン/オフ) されます。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いです。

マルチセンスコンバーター (MSC)

用語集

マルチセンスコンバーターは、レシオメトリックセンシングテクノロジーとしても知られる第 5 世代 CAPSENSE™ テクノロジーで使用されるアナログ-デジタルコンバーターです。

相互容量

ある電極 (例えば、TX) と他の電極 (例えば RX) 間の静電容量は相互容量として知られています。

負のノイズしきい値

負の方向に出るスプリアス信号から通常のノイズを識別するために使用されるしきい値です。このパラメーターは、低ベースラインリセットパラメーターとともに使用されます。

raw カウントが負のノイズしきい値を超えない (すなわち、ベースラインと raw カウントの差 (ベースライン - raw カウント) が負のノイズしきい値未満である) 限り、ベースラインは raw カウントの変化を追跡するために更新されます。

負の方向でスプリアス信号をトリガーする可能性があるシナリオは次のとおりです。電源投入時にセンサーに指が触れる場合、センサーの近くに配置される金属の物体を除去する場合、耐液性のある CAPSENSE™ 製品の水分を除去する場合、および他の急激な環境変化がある場合です。

ノイズ (CAPSENSE™ ノイズ)

センサーがオフ状態 (タッチなし) のときにピークツーピーク カウントとして測定される raw カウントの変化です。

ノイズしきい値

センサー用にノイズから信号を識別するために使用されるパラメーターです。(raw カウント - ベースライン) の差分がノイズしきい値を超える場合、おそらく有効な信号を示します。差分がノイズしきい値に下回る場合、raw カウントはノイズのみを含みます。

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材料です。センサーがある 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配置されます。製品の筐体は常にオーバーレイになります。

寄生容量 (C_p)

寄生容量は PCB 配線、センサーパッド、ビアおよびエアギャップによって与えられるセンサー電極の固有容量です。寄生容量は CSD の感度を減らすため、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 of 物体の存在を検知できるセンサーです。

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を含むウィジェットです。

raw カウント

センサーの物理的静電容量を示す CAPSENSE™ ハードウェア ブロックの未処理デジタル カウント出力です。

リフレッシュ間隔

センサーの 2 回の連続スキャンの間の時間です。

スキャン分解能

用語集

CSD ブロックによって生成される raw カウントの分解能 (ビット数) です。

スキャン時間

センサーのスキャンを完了するために要する時間です。

自己容量

回路のグラウンドと電極間の静電容量です。

感度

センサー静電容量の変化に応じる raw カウントの変化であり、(単位: カウント/pF) で表します。センサーの感度は基板レイアウト、オーバーレイ特性、センシング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 センシング方式用のスイッチト キャパシタ回路のフロント エンドを実装するために使用されるクロックソースです。

センサー

[静電容量センサー](#)を参照してください。

センサー自動リセット

システム故障の際、または金属物体がセンサーの近くに連続的に存在する際に、センサーが誤ったタッチ状態を無期限に報告してしまうことを防ぐための設定です。

センサー自動リセット機能が有効になった場合、ベースラインは差分カウントがノイズしきい値を超えても常に更新されます。このように、センサーが無期限のオン状態を報告しないようにします。センサー自動リセットが無効なとき、ベースラインは差分カウントがノイズしきい値を下回った場合にのみ更新されます。

センサー連動

[連動センサー](#)を参照してください。

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 ブロックからの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。

シールド タンク コンデンサ (C_{SH})

高い寄生容量を持つ広いシールド層がある場合、CSD シールドの駆動能力を強化するために使用されるオプションの外部コンデンサ (C_{SH} タンク コンデンサ) です。

信号 (CAPSENSE™ 信号)

差分カウントは信号とも呼ばれます。差分カウントを参照してください。

信号対ノイズ比 (SNR)

タッチしたときのセンサーの信号とタッチしないときのセンサーのノイズ信号との比率です。

スライダー分解能

スライダーが分解された指の位置の総数を示すパラメーターです。

用語集

SmartSense™ 自動チューニング

設計フェーズの後で最適性能のために、センシングパラメーターを自動的にセットし、システム、製造および環境変化に対し連続的に補正する CAPSENSE™ アルゴリズムです。

タッチパッド

特定の水平と垂直な様式で配置された複数のセンサーから成り、タッチの X および Y 位置を検出するウィジェットです。

トラックパッド

[タッチパッド](#)を参照してください。

チューニング

CAPSENSE™ の動作に必要なさまざまなハードウェアおよびソフトウェアまたはしきい値パラメーターの最適値を決定するプロセスです。

V_{REF}

PSoC™ 内にあるプログラマブル電圧リファレンスブロックであり、CAPSENSE™ および ADC の動作に使用されます。

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です。ボタン、近接センサー、リニアスライダ、ラジアルスライダ、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たウィジェットです。

改訂履歴

改訂履歴

Document version	Date of release	Description of changes
**	2013-11-01	これは英語版 001-85951 Rev. ** を翻訳した日本語版 001-89548 Rev. ** です。
*A	2015-11-20	これは英語版 001-85951 Rev. *L を翻訳した日本語版 001-89548 Rev. *A です。
*B	2016-05-17	これは英語版 001-85951 Rev. *O を翻訳した日本語版 001-89548 Rev. *B です。
*C	2017-04-17	ロゴと著作権を更新しました。
*D	2020-01-15	これは英語版 001-85951 Rev. *W を翻訳した日本語版 001-89548 Rev. *D です。
*E	2021-01-25	これは英語版 001-85951 Rev. *Y を翻訳した日本語版 001-89548 Rev. *E です。
*F	2021-06-16	テンプレートの変更を実施。 これは英語版 001-85951 Rev. *Z を翻訳した日本語版 001-89548 Rev. *F です。
*G	2022-03-17	これは英語版 001-85951 Rev. AA を翻訳した日本語版 001-89548 Rev. *G です。

Trademarks

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

Edition 2022-03-17

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2022 Infineon Technologies AG.

All Rights Reserved.

Do you have a question about this document?

Go to www.infineon.com/support

Document reference

001-89548 Rev. *G

重要事項

本文書に記載された情報は、いかなる場合も、条件または特性の保証とみなされるものではありません（「品質の保証」）。本文に記載された一切の事例、手引き、もしくは一般的な価値、および/または本製品の用途に関する一切の情報に関し、インフィニオンテクノロジーズ（以下、「インフィニオン」）はここに、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。

さらに、本文書に記載された一切の情報は、お客様の用途におけるお客様の製品およびインフィニオン製品の一切の使用に関し、本文書に記載された義務ならびに一切の関連する法的要件、規範、および基準をお客様が遵守することを条件としています。

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。

本製品、技術、納品条件、および価格についての詳しい情報は、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 (www.infineon.com)。

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。