PSOC™ 3 および PSOC™ 5LP CAPSENSE™ デザインガイド

本書について

適用範囲と目的

本書は、PSOC™ 3 および PSOC™ 5LP ファミリーのデバイスを使用した CAPSENSE™ 設計のガイドを提供します。

対象者

これは、PSOC™ 3 および PSOC™ 5LP ファミリーのデバイスを使用してアプリケーションを設計する、静電容量センシング技術に精通するエンジニアを対象とします。

目次

目次

本書について 1
目次 2
1 はじめに 5
1.1 インフィニオン CAPSENSE™ の文書体系 5
1.2 PSOC™ 3 および PSOC™ 5LP デバイスの特長 8
1.3 本書の表記法 9
2 CAPSENSE™ 技術 10
2.1 CAPSENSE™ 基礎 10
2.2 静電容量の変換 11
2.2.1 容量性シグマデルタ (CSD) 12
2.2.2 CSD の実装 13
2.3 SmartSense 自動チューニング 13
3 PSOC™ 3 および PSOC™ 5LP の CAPSENSE™ 14
3.1 CSD の実装 14
3.2 CAPSENSE™ 独自の機能 15
3.2.1 2 チャネル デザイン 15
3.2.2 耐水性デザイン 15
3.2.2.1 SIO ピン 16
3.2.3 電流ソース方式 16
3.2.3.1 IDAC ソース方式 17
3.2.3.2 IDAC シンク方式 18
3.2.3.3 外部抵抗方式 19
3.2.4 チューニング 20
3.2.4.1 自動チューニング機能 20
3.2.4.2 手動チューニング機能 21
3.2.4.3 ノンブロッキング アーキテクチャ 22
3.3 CAPSENSE™ PLUS 22
4 CAPSENSE™ 設計ツール 23
4.1 PSOC™ Creator 23
4.2 ハードウェアキット 24
4.2.1 PSOC™ 3 および PSOC™ 5LP 開発キット 24
4.2.2 ユニバーサル CAPSENSE™ モジュール ボード 24
4.2.3 PSOC™ CAPSENSE™ 拡張ボード キット 24
5 CapSense_CSD コンポーネント 25
5.1 パラメーターのまとめ 26
5.2 グローバルアレイ 28
5.2.1 raw カウント 28
5.2.2 ベースライン 28
5.2.3 差分カウント 29
5.2.4 センサー状態 29
5.3 高レベル パラメーター 29
5.3.1 指閾値 30
5.3.2 ヒステリシス 30
5.3.3 デバウンス 30
5.3.4 ノイズ閾値 31
5.3.5 負のノイズ閾値および低ベースラインリセット 31
5.3.6 センサー自動リセット 32

目次

5.3.7 ウィジェット分解能 33
5.3.8 フィルタの選択 34
5.3.8.1 raw データノイズフィルタ 34
5.3.8.2 位置ノイズフィルタ 35
5.3.9 高レベルパラメーターの推奨事項 35
5.4 低レベルパラメーター 36
5.4.1 クロック設定 36
5.4.1.1 スキャンクロック 36
5.4.1.2 クロック入力のイネーブル 37
5.4.2 アナログスイッチ分周器 38
5.4.2.1 アナログスイッチ分周器 (プリスケーラー) 38
5.4.2.2 複数のアナログスイッチ分周器 39
5.4.2.3 アナログスイッチドライブソース 40
5.4.3 擬似乱数シーケンス (RPS) 41
5.4.4 スキャン分解能 42
5.4.5 IDAC 電流 43
5.4.5.1 IDAC 値 43
5.4.5.2 IDAC 範囲 44
5.4.6 スキャン速度 44
5.4.7 デジタルリソースの実装 46
5.4.8 電流ソース 47
5.4.8.1 ブリード抵抗数 48
5.4.9 電圧基準ソース 49
5.4.10 シールド電極およびガードセンサー 50
5.4.10.1 シールド 50
5.4.10.2 非アクティブなセンサー接続 50
5.4.10.3 ガードセンサー 50
5.4.10.4 シールド電極用に SIO ピンの使用 51
5.4.10.5 複数のシールド電極ピン 52
5.4.10.6 耐水性および検知 52
5.5 ウィジェットのコンフィギュレーション 53
5.5.1 ウィジェットの追加 53
5.5.2 センサーエレメント数 53
5.5.3 API 分解能 53
5.5.4 近接センサー 54
5.5.4.1 専用センサーエレメント数 55
5.6 チューニング方法 56
5.6.1 感度 57
5.7 チャネル数 57
5.7.1 チャネル 1/チャネル 0 への変更 58
5.7.2 2 チャネル設計のピン割り当て 59
5.8 チューナー GUI 60
5.8.1 設定 60
6 CAPSENSE™ 性能のチューニング 65
6.1 基本事項 65
6.1.1 信号、ノイズおよび SNR 65
6.1.1.1 CAPSENSE™ 信号 65
6.1.1.2 CAPSENSE™ ノイズ 65
6.1.1.3 CAPSENSE™ SNR 66
6.1.2 充電/放電速度 66

目次

6.2 マニュアル調整 67
6.3 SmartSense 自動チューニング 69
6.3.1 ガイドライン 69
6.3.2 パラメーター設定 69
6.3.2.1 センサー自動リセット 69
6.3.2.2 デバウンス 69
6.3.2.3 変調器コンデンサピン 69
6.3.2.4 感度レベル 69
7 設計上の注意事項 71
7.1 ソフトウェアのフィルタリング 71
7.2 消費電力 72
7.2.1 スリープスキャン方式 73
7.2.2 平均消費電力の測定 74
7.2.2.1 アクティブ電流 74
7.2.2.2 アクティブ時間 74
7.2.2.3 スリープ電流 75
7.2.2.4 スリープ時間 75
7.2.2.5 平均消費電力の計算 75
7.3 応答時間 76
7.3.1 スリープスキャンモード 76
7.3.2 長いバックグラウンドループ 76
7.3.3 デバウンス 77
7.3.4 フィルタによる遅延 78
7.3.5 割込み優先順位 79
7.4 ピン割り当て 80
7.4.1 オペアンプ出力ピン 80
7.4.2 2 チャネル設計のピン割り当て 80
7.4.3 C_{MOD} ピン割り当て 80
7.5 プリント基板レイアウトガイドライン 80
8 用語集 81
9 参考資料 87
改訂履歴 89
免責事項 90

1 はじめに

本書は、PSOC™ 3 および PSOC™ 5LP ファミリーのデバイスを使用して CAPSENSE™ アプリケーションを設計するためのガイダンスを提供します。これは、容量性センシング技術に精通しており、アプリケーションに PSOC™ 3 および PSOC™ 5LP ファミリーのデバイスを選択した設計エンジニアを対象とします。CAPSENSE™ 技術の完全な理解については、CAPSENSE™ 入門を参照してください。

1.1 インフィニオン CAPSENSE™ の文書体系

Figure 1 および Table 1 は、インフィニオン CAPSENSE™ 文書体系の要約です。これらのリソースにより、CAPSENSE™ 製品設計を完了する際に必要な情報に迅速にアクセスできます。Figure 1 に、静電容量センシングの標準的な製品設計サイクルを示します。本ガイドでは、緑色でハイライト表示した項目を説明します。Table 1 には Figure 1 で番号付けされた各タスクをサポートする文書へのリンクが提供されています。

1 はじめに

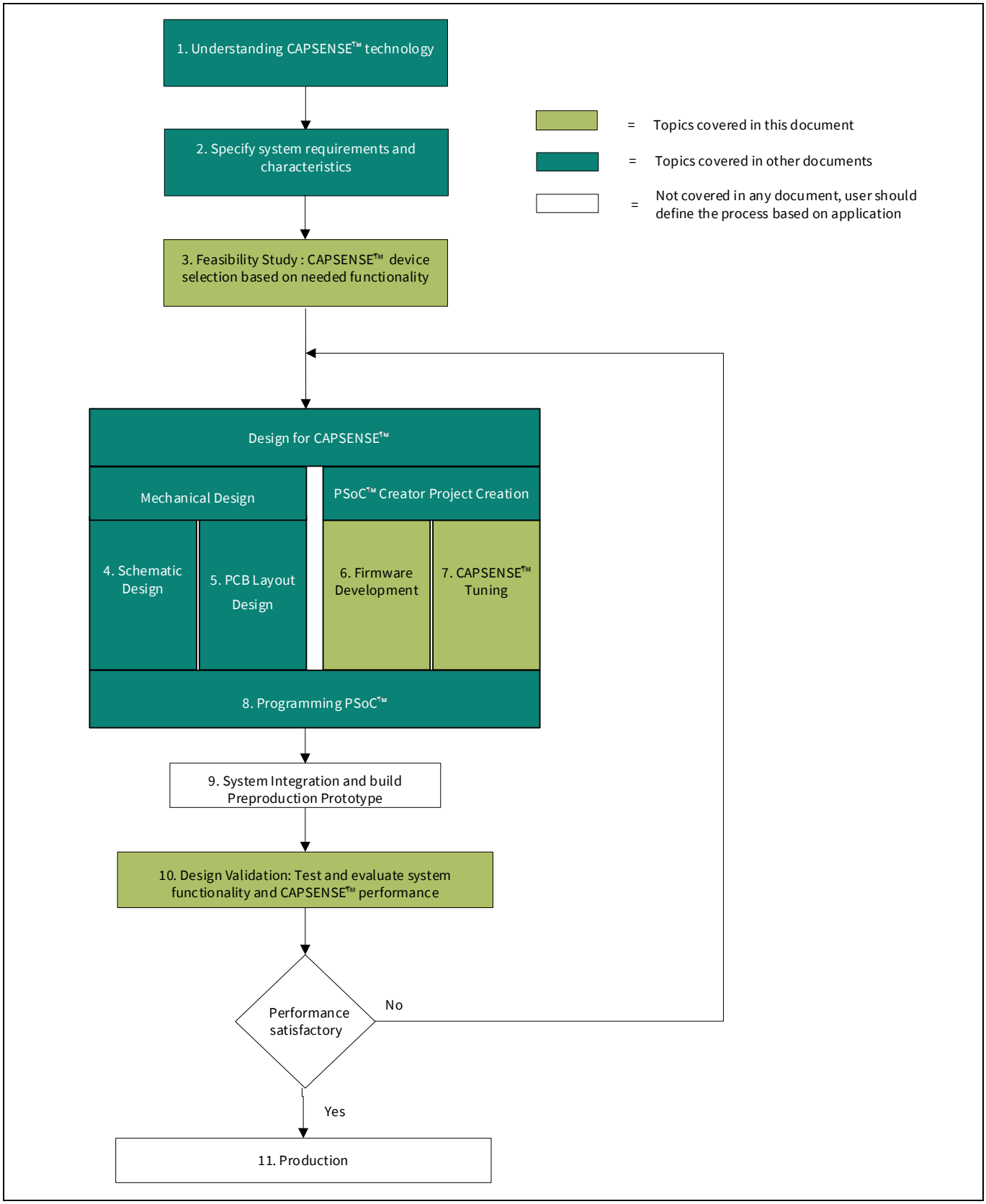

Figure 1　標準的な CAPSENSE™ 製品設計フロー

1 はじめに

Table 1 **Figure 1** 中に番号付けされた設計タスクをサポートするインフィニオンのドキュメント

番号付けされた設計タスク	インフィニオン CAPSENSE™用サポートドキュメント		
	名称	セクション	説明
1	CAPSENSE™ 入門	2	CAPSENSE™ 動作の詳細な理論
2	CAPSENSE™ 入門	4 および 5	デバイスの特長を説明し、デバイスを選択する手助けをする
	PSOC™ 3 データシート PSOC™ 5LP データシート	-	
3	CAPSENSE™ 入門	4 および 5	デバイスの特長を説明し、デバイスを選択する手助けをする
	本書	3	
4	CAPSENSE™ 入門	3	C_{MOD} にピン割り当ておよび値などの情報を提供
	本書	7	
5	CAPSENSE™ 入門	3	PCB レイアウト ガイドラインを提供
6	PSOC™ Creator ヘルプ トピック (PSOC™ Creator IDE の Help タブにある)	-	PSOC™ Creator ヘルプ トピックは PSOC™ Creator IDE を使用するガイドラインを提供 本書および CAPSENSE™_CSD コンポーネント データシートは CAPSENSE™ アプリケーションを開発するファームウェア ガイドラインを提供
	本書	4, 5, 6, および 7	
	CapSense_CSD コンポーネント データシート	“Application programming interface”	
7	本書	6	CAPSENSE™ システムを調整する方法を説明
8	MiniProg3 ユーザガイド	-	PSOC™ 3 および PSOC™ 5LP のプログラミングを詳しく説明
	AN61290 - PSOC™ 3/PSOC™ 5LP Hardware Design Considerations	“Programming and Debugging”	
9	該当なし	-	-
10	CAPSENSE™ 入門	3	設計上の重要な注意事項
	本書	6 および 7	
11	該当なし	-	-

1.2 PSOC™ 3 および PSOC™ 5LP デバイスの特長

PSOC™ 3 および PSOC™ 5LP は、コンフィギュレーション可能なアナログおよびデジタル ペリフェラル機能、メモリ、マイクロコントローラーを搭載したプログラマブルな組込みシステムオンチップです。これらのデバイスは非常に柔軟性が高く、CAPSENSE™ の他に多くの機能を実装できます。主な機能は次のとおりです。

デバイスの特長

- PSOC™ 3 向け 67 MHz 8051 CPU
- PSOC™ 5LP 向け 67MHz Arm® Cortex® -M3 CPU
- PSOC™ 3 向け 最大 64KB フラッシュ、8KB SRAM、2KB EEPROM
- PSOC™ 5LP 向け 最大 256KB フラッシュ、64KB SRAM、2KB EEPROM
- 24 チャネル DMA
- 最大 72 の I/O ピン
- 低消費電力モード
  - PSOC 3 向け 1 µA のスリープ モード電流
  - PSOC 5LP 向け 2 µA のスリープ モード電流
  - PSOC 3 向け 200 nA のハイバネート モード電流
  - PSOC 5LP 向け 300 nA のハイバネート モード電流
- アナログ機能
  - 分解能が 8～20 ビットのコンフィギュレーション可能なデルタ シグマ ADC
  - 最大 4 個のコンパレータ、オペアンプ、DAC、多機能アナログ ブロック
  - 0.1% 内部バンドギャップ基準電圧
- デジタル機能
  - 最大 4 個の 16 ビットのコンフィギュレーション可能なタイマ、カウンター、PWM ブロック
  - 最大 24 個のプログラマブル ロジック デバイス (PLD) ベースのユニバーサル デジタル ブロック (UDB)
- 幅広いパッケージ: QFN、SSOP、TQFP
- 任意の GPIO で最大 46 × 16 セグメントの LCD を直接駆動
- I2C、UART、フルスピード USB、SPI、CAN 通信インターフェース

CAPSENSE™ の特長

- CAPSENSE™ ボタン、スライダー、マトリックス ボタンおよび近接センサーの組み合わせに対応
- 最大 61 個の静電容量センサーに対応し、すべての GPIO ピンで CAPSENSE™ をサポート
- ファームウェアを開発する API を統合
- 防水設計をサポート
- 2 チャネル デザイン: 2 個のセンサーを同時にスキャンするのに十分なリソース
- SmartSense 自動チューニング
  - 電源投入時と実行中のチューニング パラメーターを自動的に設定および監視
  - デザインの移植性のために、ユーザーインターフェース設計の変化に適応
  - 実行時の環境の変化を補正
  - 最小 0.1 pF の静電容量でタッチを検知

1.3 本書の表記法

表記	説明
Courier New フォント	ファイルの場所、ユーザーが入力したテキスト、ソースコードを表示: `C:\...cd\icc\`
イタリック体	ファイル名および参考資料を表示: *PSOC ™Designer User Guide* にある *sourcefile.hex* ファイルを参照してください。
[角括弧、太字]	手順としてキーボードのコマンドを示す: **[Enter]** または **[Ctrl] [C]**
File > Open	メニューパスを表示: File > Open > New Project
太字	コマンド、メニューパス、アイコン名を表示: **File** アイコンをクリックして、**Open** をクリック。
Times New Roman フォント	式を表示: 2 + 2 = 4

2 CAPSENSE™ 技術

2.1 CAPSENSE™ の原理

CAPSENSE™ は、センサーとして指定された各 I/O ピンの静電容量を測定することによって機能するタッチセンシング技術です。Figure 2 に示すように、各センサー ピンの総合静電容量は、Cx1 ～ Cxn の値を持つ等価集中コンデンサとしてモデル化できます。CAPSENSE™ 技術は、外部変調コンデンサ C_{MOD} を必要とします。C_{MOD} の詳細は、静電容量の変換 を参照してください。

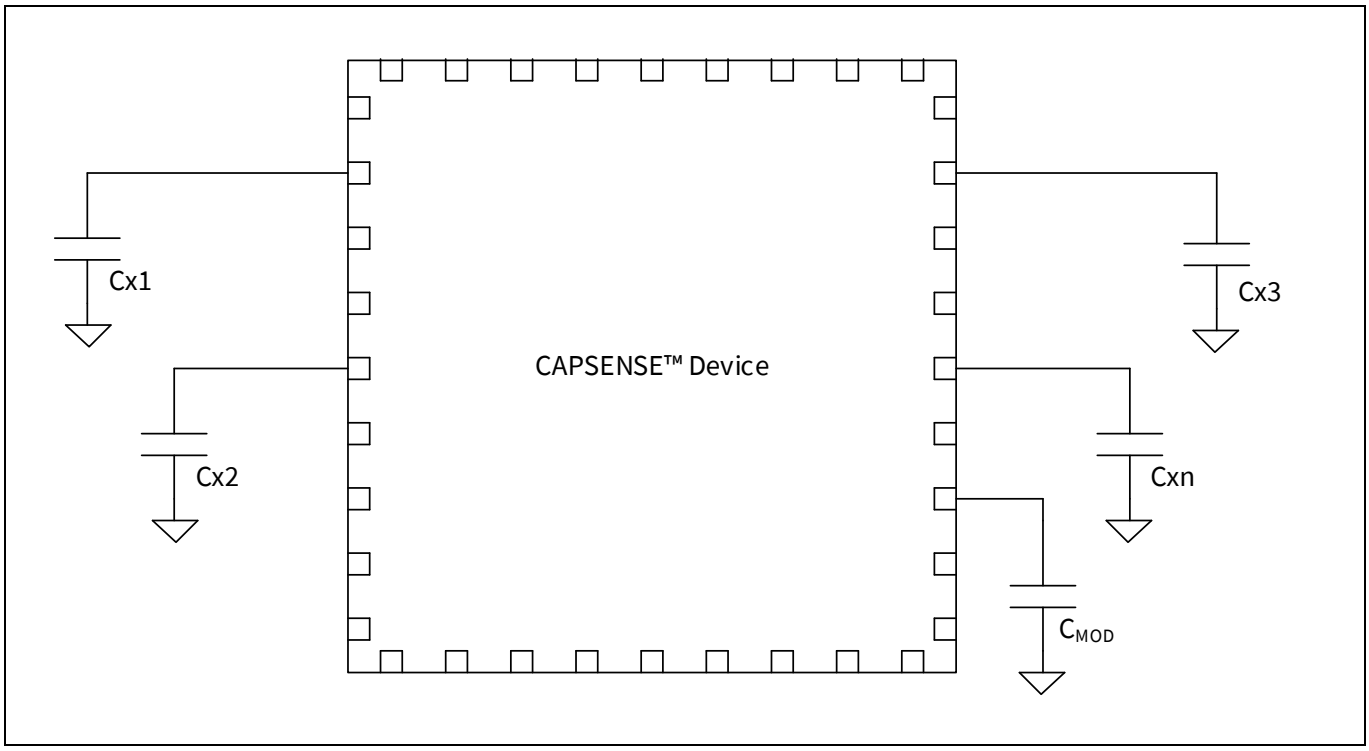

Figure 2 CAPSENSE™ デバイススキャニングセンサー Cx1～Cxn

各センサーの I/O ピンは、配線、ビア、または必要に応じてその両方でセンサーパッドに接続されます。非導電性オーバーレイがセンサーパッドを覆うのに必要であり、システムのタッチ インターフェースを構成しています。指がオーバーレイに接触すると、人体の伝導性と大きさにより、グランドに接続された導体板がセンサーパッドと平行に置かれるのと同じ状況になります。これを Figure 3 に示します。この配置は平行板コンデンサを構成し、その静電容量は以下の式によって与えられます。

$$C_F = \frac{\varepsilon_0 \varepsilon_r A}{D}$$

Equation 1

ここで、

C_F = センサーを覆うオーバーレイに接触する指により生じた静電容量

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

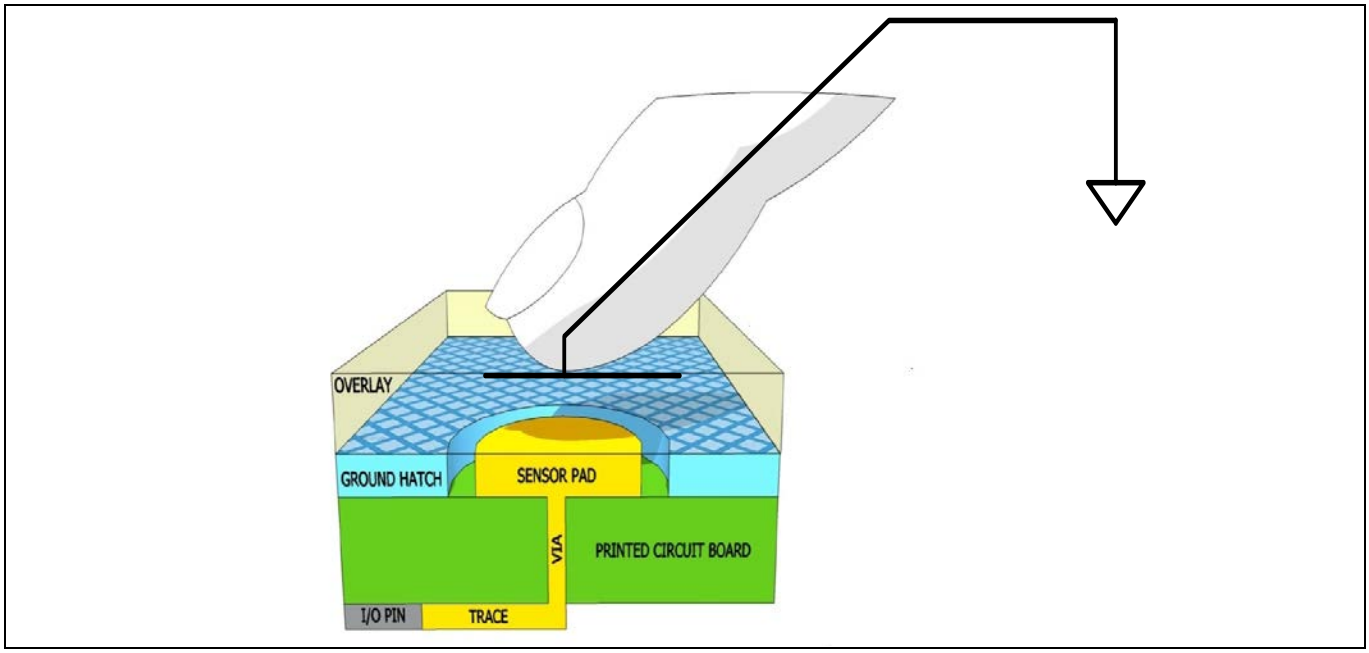

Figure 3　CAPSENSE™ システム等価モデル

指がオーバーレイに接触しなくても、センサーの入力ピンには寄生容量 (C_P) があります。C_P は、CAPSENSE™ コントローラー内部の寄生容量と電界の組み合わせの結果です。この電界は、センサーパッド、配線、ビア、およびシステム内のその他の導体 (グランドプレーン、その他の配線、製品のシャーシまたは封入物内の金属など) 間のカップリングにより生じた電界です。C_P および C_F は、両者がセンサーピンとグランドの間で接続されているため、互いに並列です。

指がセンサーに触れていない場合:

$$C_X = C_P$$

Equation 2

指がセンサーに触れている場合:

$$C_X = C_P + C_F$$

Equation 3

一般的に、C_P は C_F よりも何十倍か大きい値です。C_P は通常 6～15 pF の範囲ですが、極端な場合は 45 pF まで高くなることもあります。C_F は通常 0.1～0.4 pF の範囲です。C_P の大きさは、CAPSENSE™ システムのチューニング時に重要です。最適な性能を得るには、C_P をできる限り低くする必要があります。

2.2　静電容量の変換

CAPSENSE™ デバイスは、各 Cx の大きさをデジタル カウントに変換します。このデジタルカウントは、センサーパッド上またはその近くの指の存在を検出するために保存および処理されます。

2.2.1 容量性シグマデルタ (CSD)

容量性シグマデルタ (CSD) は、センサーの静電容量をデジタルカウントに変換する方式の 1 つです。CSD 方式は他のセンシング方式を凌ぎます。Figure 4 に CSD 方式のブロック図を示します。

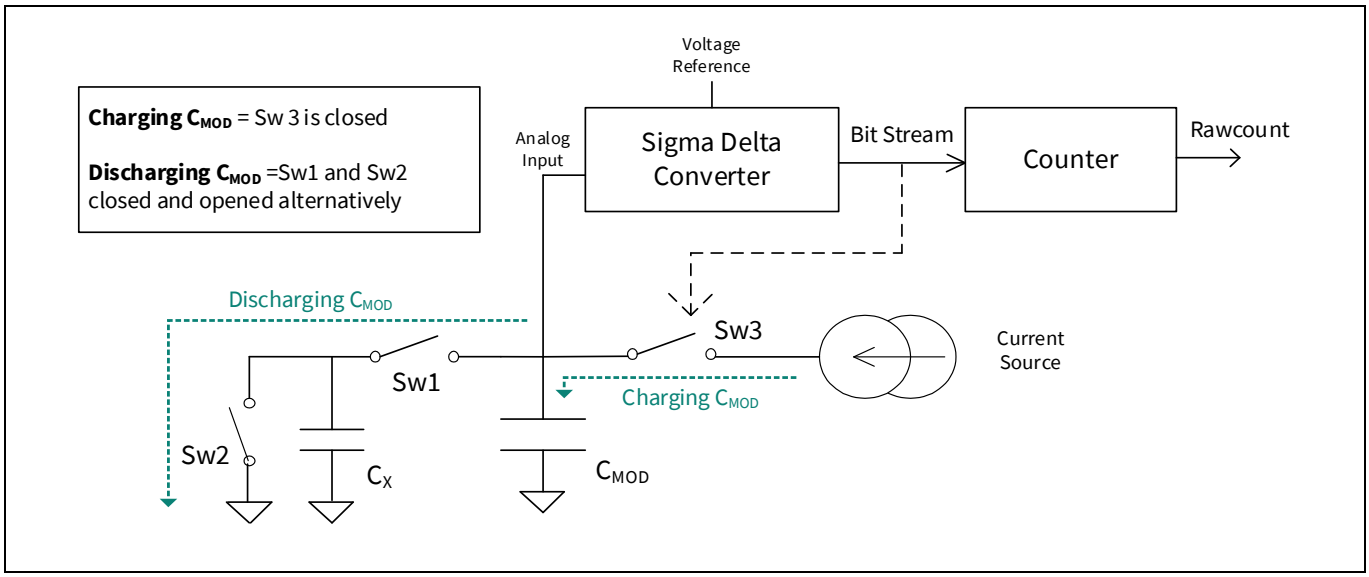

Figure 4 CAPSENSE™ CSD ブロック図

CSD 方式には、変調コンデンサ (C_{MOD}) と呼ばれる大きな積分コンデンサが必要です。センサーコンデンサ C_X は、スイッチ Sw1 と Sw2 に接続され、スイッチト コンデンサ ブロックを形成します。Sw1 は C_X を C_{MOD} に、Sw2 は C_X をグランドに接続します。Sw1 と Sw2 は交互に開閉し (同時に閉じない)、C_{MOD} の放電パスを提供します。

シグマ デルタ コンバータは、C_X の静電容量をデジタルカウントに変換します。定電流ソースはスイッチ Sw3 を介して C_{MOD} に接続されます。Sw3 が閉じると、電流ソースは C_{MOD} を充電します。C_{MOD} は入力としてシグマデルタコンバータに接続されます。シグマデルタコンバータはその入力に応じて、Sw3 を制御して C_{MOD} の平均電圧を基準電圧に維持するようにします。シグマ デルタ コントローラーの出力は、Sw3 のデューティサイクルを示すビットストリームです。

Sw3 のデューティサイクルは C_X の静電容量に正比例します。例えば、C_X の値が高くなると、C_{MOD} の放電電流が増加します。基準電圧で C_{MOD} 電圧を維持するように、シグマデルタコンバータは Sw3 のデューティサイクルを増加します。

ビットストリームは、raw カウントとして知られるデジタル値に変換されます。高レベルのアルゴリズムによって raw カウントは解釈され、センサーの状態を判断します。指がセンサーに触れると、C_X が C_F の分だけ増加し、raw カウントがそれに比例して増加します。定常状態の raw カウントの動きと事前に設定された閾値を比較することで、センサーがオン (接触) 状態であるかオフ (非接触) 状態であるかを高レベルアルゴリズムにより判定できます。Figure 5 は、センサーを指で触れてから離すまで連続スキャンした raw カウントのプロットを示します。

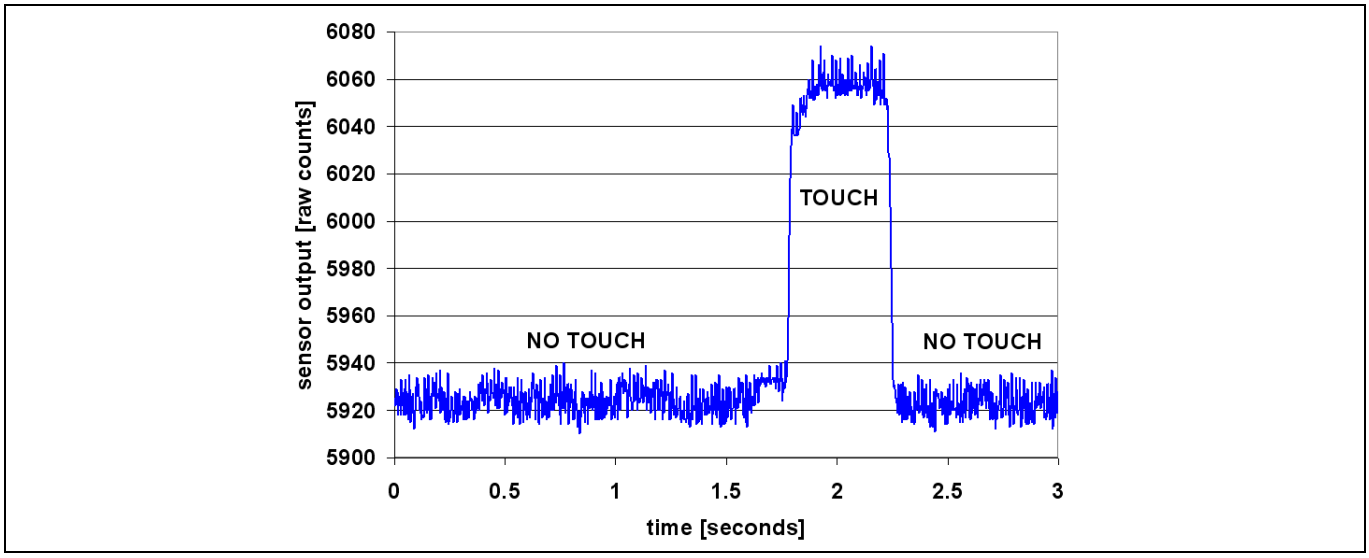

Figure 5 raw カウントと時間

インフィニオンの CSD センシング方式の詳細については、PSOC™ 3 architecture reference manual および PSOC™ 5LP architecture reference manual を参照してください。

2.2.2 CSD の実装

CSD 方式を実装する方法は多くあります。

IDAC ソース方式: Figure 5 に示すように、IDAC は C_{MOD} を充電し、センサーは C_{MOD} を放電します。

IDAC シンク方式: IDAC は C_{MOD} を放電し、センサーは C_{MOD} を充電します。

外部抵抗方式: 外部抵抗は C_{MOD} を放電し、センサーは C_{MOD} を充電します。

上記の CSD 方式の実装方法は 電流ソース方式 に記載されています。

2.3 SmartSense 自動チューニング

CAPSENSE™ システムを構成するハードウェアとフォームウェアには、CAPSENSE™ システムの動作を決めるいくつかのパラメーターがあります。これらのパラメーターのチューニングは、適切なシステムの動作と快適なユーザーの使用体験のために重要です。残念ながら、チューニングは反復プロセスなので時間がかかります。一般的な開発サイクルでは、インターフェースは初期の設計段階、システム統合中および増産前にチューニングされます。

SmartSense 自動チューニングはインフィニオンの先進技術です。SmartSense は、広範囲のアプリケーションでシステム性能を自動的に最適化する洗練されたアルゴリズムです。それは使いやすく、試作や製造段階で手動チューニングを無くすことで、設計のサイクル時間を短縮します。SmartSense 自動チューニングは電源投入時に各 CAPSENSE™ ボタンを自動的に調整し、使用中の最適なボタン操作を維持します。これは、PCB 基板とオーバーレイの製造工程のばらつきに適応しており、LCD インバーター、AC ライン、スイッチング電源などのノイズ源を自動的に調整してノイズを取り除きます。

SmartSense 自動チューニングは SmartSense 自動チューニング で詳しく説明されています。

3 PSOC™ 3 および PSOC™ 5LP における CAPSENSE™

PSOC™ 3 および PSOC™ 5LP はプログラム可能なデバイスであり、豊富なアナログおよびデジタル リソースが用意されています。これらのデバイスは、多数のユニークな機能を持った、CSD 方式を用いた CAPSENSE™ を実装します。これらのデバイスは柔軟性があり、リソースが豊富なため、CAPSENSE™ やその他多くのシステム機能を実行できます。

3.1 CSD の実装

CSD 方式の詳細な説明は、静電容量の変換 を 参照してください。Figure 6 に、PSOC™ 3 および PSOC™ 5LP デバイスで CSD 方式をどのように実装するかを示します。これらのデバイスには最大 4 つの DAC リソースがあります。CAPSENSE™ は、これらのいずれかを使用して電流ソース (IDAC) を実装します。スイッチは、非重複クロック (スイッチングクロック) によって制御されます。アナログバス (AMUXBUS) は、C_{MOD}、C_X、IDAC、およびシグマ デルタ コンバータの入力を接続します。シグマ デルタ コンバータは、IDAC を制御して C_{MOD} 電圧が基準電圧 (Vref) に近似するようにします。シグマ デルタ コンバータは、カウンターにビットストリームを出力します。カウンターは raw カウントを出力します。CPU は raw カウントを処理してセンサーの状態を判断します。

Figure 6 に CSD の IDAC ソースの実装方式を示します。その他の方式は 電流ソース方式 を参照してください。

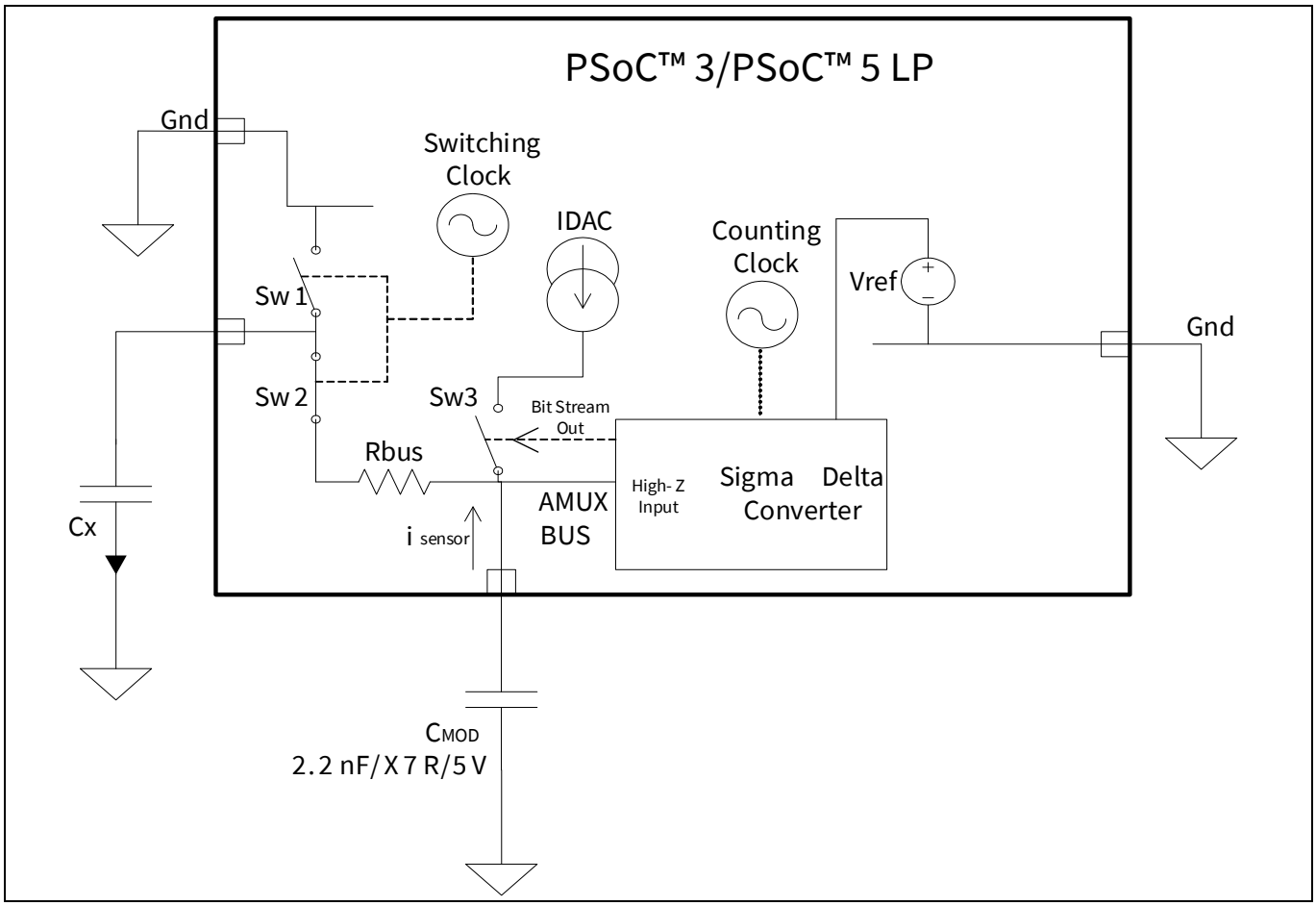

Figure 6 CSD の実装 (IDAC ソース方式)

3.2 CAPSENSE™ 独自の特長

3.2.1 2 チャネル デザイン

PSOC™ 3 および PSOC™ 5LP デバイスは、2 個のセンサーを同時にスキャンすることが可能です。これにより、スキャン時間は半減されます。スキャン時間を短縮することにより、ボタン オン/オフ検出の応答時間が増え、平均消費電力を削減できます。

2 チャネルを使用すると、1 チャネルのリソースが 2 倍必要となるので、20 個以上のセンサーを持っている設計にのみ推奨されています。チャネル数 では、2 チャネルデザインを選択する方法が説明されています。Table 2 に、1 チャネルデザインと 2 チャネルデザインに必要なリソースを比較します。

Table 2 1 チャネルと 2 チャネル デザインのリソース比較

リソースタイプ	1 チャネル	2 チャネル
アナログリソース [1]	AMUXBUS 1 本 または 2 本 コンパレータ 1 個 DAC 1 個	AMUXBUS 2 本 コンパレータ 2 個 DAC 2 個
デジタルリソース [2]	データパス 4 本 マクロセル 19 個	データパス 6 本 マクロセル 31 個
外部コンポーネント [3]	C_{MOD} コンデンサ 1 個 ブリード抵抗 1 セット	C_{MOD} コンデンサ 2 個 ブリード抵抗 2 セット

3.2.2 耐水性デザイン

一部の CAPSENSE™静電容量タッチ センシング システムは、水が存在する場では信頼性の高い動作を必要とします。家電製品や車載アプリケーション、産業用アプリケーションは、水や氷、湿度の変化を伴う環境で動作する必要があるシステムの例です。そのようなアプリケーションには、シールド電極およびガードセンサーで堅牢なタッチセンシングを提供することが可能です。

水滴や湿気の耐性が必要な用途の場合は、シールド電極を使用する必要があります。Figure 7 に示すように、シールド電極はセンサーの周りを囲んでいます。シールド電極はデバイスの I/O ピンに接続され、センサーピンと同じスイッチング信号で駆動されます。同じ信号でシールド電極とセンサーの両方を駆動すると、それらの間の静電容量がゼロになります。つまり、センサーとシールドを部分的に覆っている水膜や水滴が効果的に除去されます。シールド電極はセンサーの C_P も低減します。必要な I/O ピン数は、基板サイズとシールド面積によって異なります。シールド電極の占める面積が大きい場合は、複数の I/O ピンで駆動する必要があります。

アプリケーションにおいて接触面の水流に対する耐性を必要とする場合、シールド電極とともにガードセンサーを使用してください。ガードセンサーはタッチセンシング領域の全体を被う必要があります。Figure 7 に示すように、ガードセンサーは通常、タッチセンシング領域の境界を覆います。ガードセンサーは、他のセンサーと同じ方法で CAPSENSE™デバイスによってスキャンされます。ガードセンサー上

[1] 左 AMUXBUS と右 AMUXBUS があります。1 チャネル デザインは 1 本だけの AMUXBUS が必要で、すべてのセンサーがチップの片側に配置されています。2 チャネル デザインはセンサーをチップの両側に配置し、左 AMUXBUS と右 AMUXBUS を短絡させます。IDAC ソース方式または IDAC シンク方式が選ばれた場合、少なくとも 1 個の DAC が必要です。電流ソース方式 を参照してください。

[2] データパスとマクロセルは、ユニバーサル デジタル ブロック (UDB) のコンポーネントのブロックです。PSOC 3 および PSOC 5LP デバイスには、最大 24 の UDB ブロックがあります。

[3] 外部抵抗方式が選択された場合、ブリード抵抗が必要になります。電流ソース方式 を参照してください。

に水が存在すると、それがアクティブになり、誤った指タッチを検出することを防ぐために他のセンサーのスキャンを無効にします。

シールド電極およびガードセンサー では、シールド電極とガードセンサーを有効にする方法を説明します。

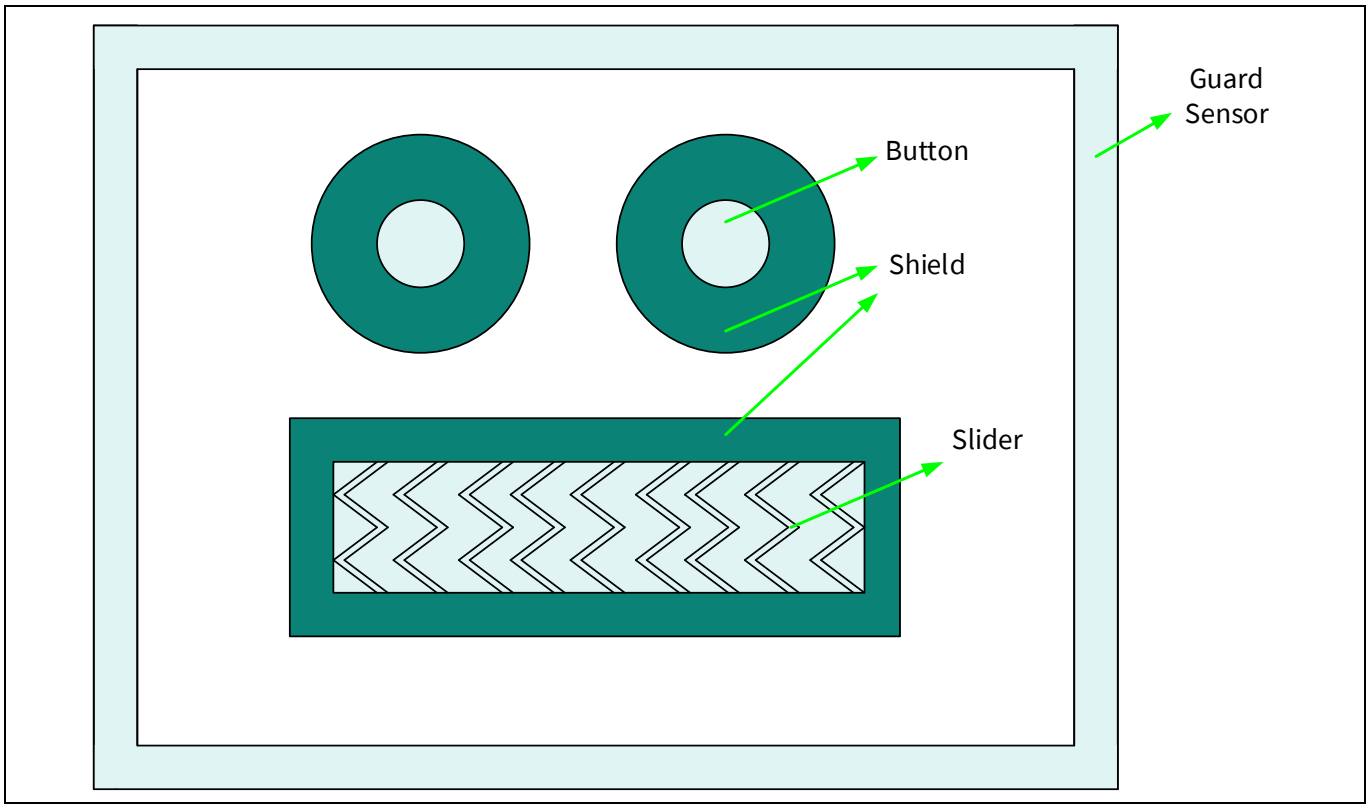

Figure 7　シールド電極およびガードセンサー

3.2.2.1　SIO ピン

PSOC™ 3 および PSOC™ 5LP デバイスには、プログラマブルな「論理 HIGH」レベルという機能を備えた特別な入出力 (SIO) ピンがあります。つまり、SIO の論理 HIGH レベルは V_{DD} に固定されず、ユーザー定義の電圧にプログラムできます。

この機能は、SIO ピンがシールド電極として使用される場合に重要です。シールド電極は、その信号が CAPSENSE™ センサーの信号と一致する場合に最高の結果を生み出します。SIO ピンを使用することにより、CAPSENSE™ センサーに一致する論理 HIGH レベルを選択できます。

3.2.3　電流ソース方式

PSOC™ 3 および PSOC™ 5LP デバイスは 3 つの CSD 実装方式の中で 1 つを選択できます。デザインにおいて最高の性能を生み出す方式を判定する際、ソース要件、ノイズ感受性、シールド電極の要件などいくつかの要素があります。これらのすべての方式では、Vref の初期設定値はバンドギャップ電圧 1.024 V に相当します。

Table 3 に 3 つの方式の比較を示します。PSOC 3 および PSOC 5LP デバイスには、型番に応じて最大 4 個の DAC があります。設計上の他の機能はすべての DAC を使用する場合、CAPSENSE™ 用に外部抵抗方式を使用することが可能です。電流ソース は電流ソース方式の選択方法について説明します。

3.2.3.1 IDAC ソース方式

この設定では、CAPSENSE™ センサーは C_{MOD} と GND 間を連続的に切り替え、C_{MOD} を放電します。IDAC はソースモードに設定され、C_{MOD} 電圧が Vref 以下になるとオンになります。Figure 8 に、IDAC ソース方式のブロック図を示します。

この方式は、センサーの電圧振幅が小さい (Vref と GND の間) ため、指伝導ノイズの影響を受けやすいものです。VDAC で Vref を増加させると、ノイズ耐性が高くなりますが、追加の DAC リソースが必要となります。

IDAC ソース方式を使用した場合、SIO ピンをシールド電極に使用します。これにより、シールド電極を Vref と GND 間で切り替えることができ、センサーと同じ信号を取得することを保証します。

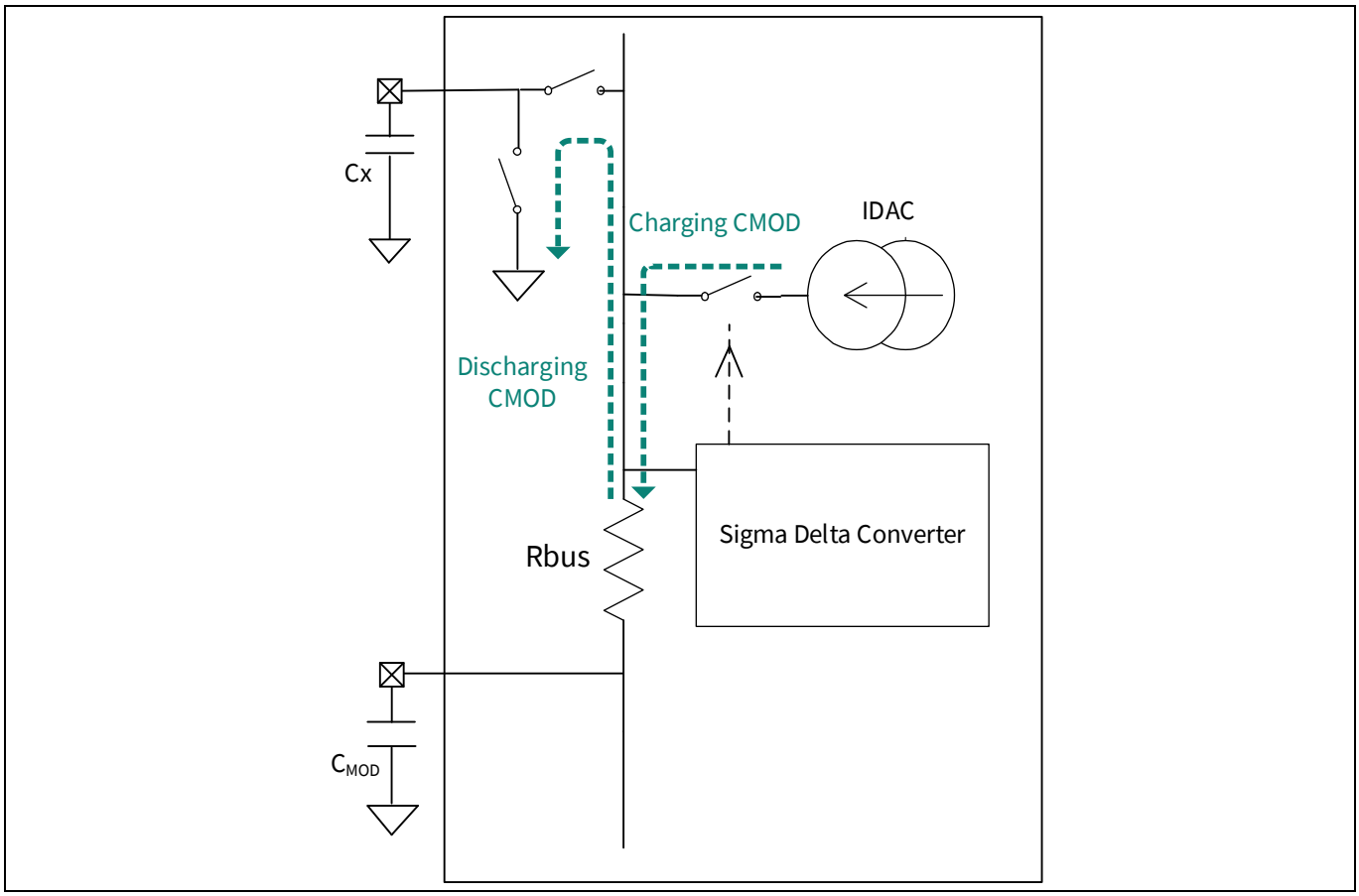

Figure 8 IDAC ソース方式

3.2.3.2 IDAC シンク方式

この設定では、CAPSENSE™ センサーは V_{DD} と C_{MOD} 間を連続的に切り替え、C_{MOD} を充電します。IDAC はシンクモードに設定され、C_{MOD} 電圧が Vref を超えるとオンになります。Figure 9 に、IDAC シンク方式のブロック図を示します。

この方式は、センサーが V_{DD} と Vref 間を切り替えるため、電源ノイズの影響を受けやすいものです。

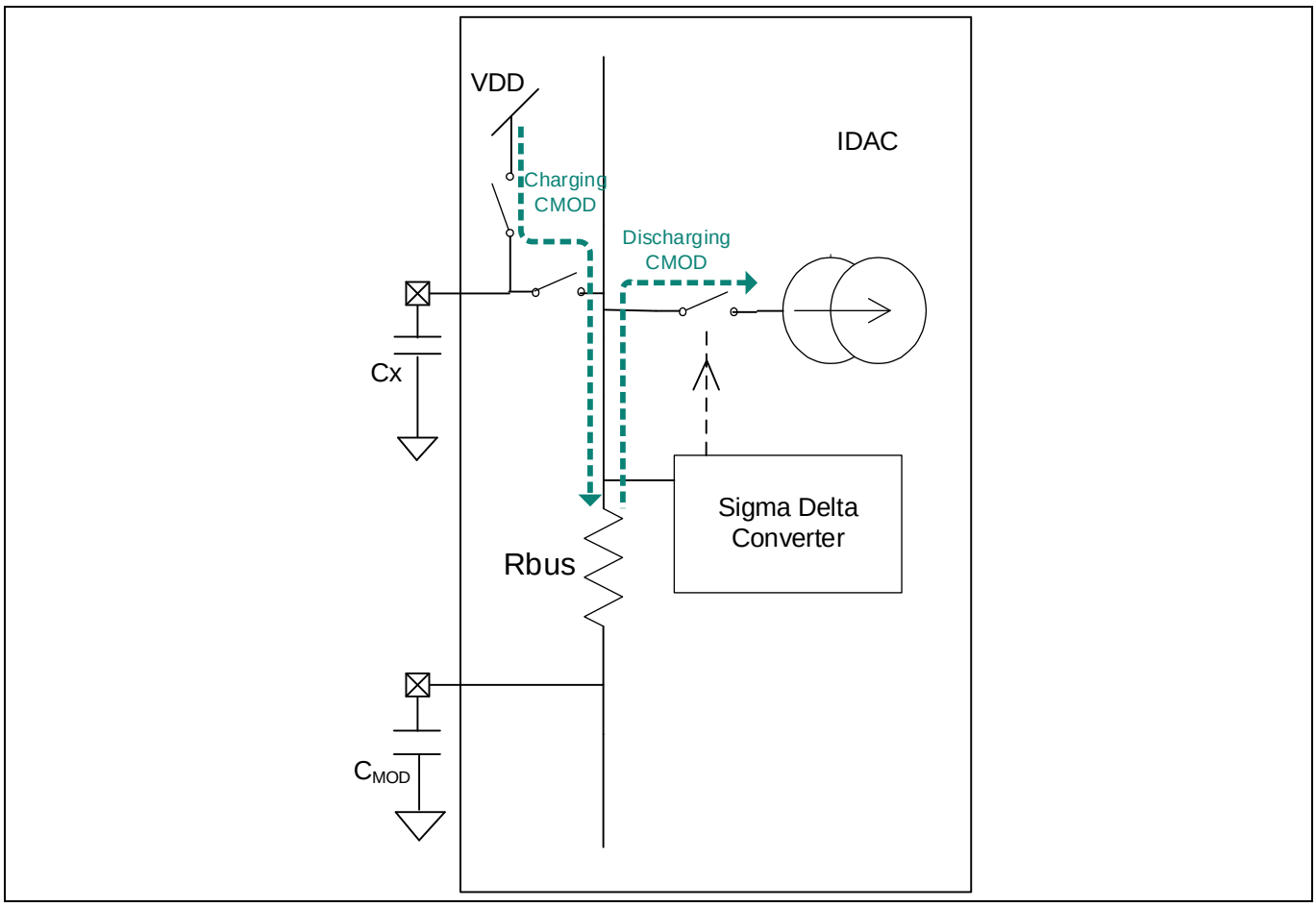

Figure 9 IDAC シンク方式

3.2.3.3 外部抵抗方式

この設定は、IDAC の代わりに外部抵抗 (ブリード抵抗) を用いて C_{MOD} を放電することを除き、IDAC シンク方式に似ています。CAPSENSE™ センサーは、V_{DD} と C_{MOD} 間を連続的に切り替え、C_{MOD} を充電します。C_{MOD} 電圧が Vref を超えると、ブリード抵抗はグランドに接続されます。Figure 10 に、外部抵抗方式のブロック図を示します。

この方式は、センサーが V_{DD} と Vref 間を切り替えるため、電源ノイズの影響を受けやすいものです。

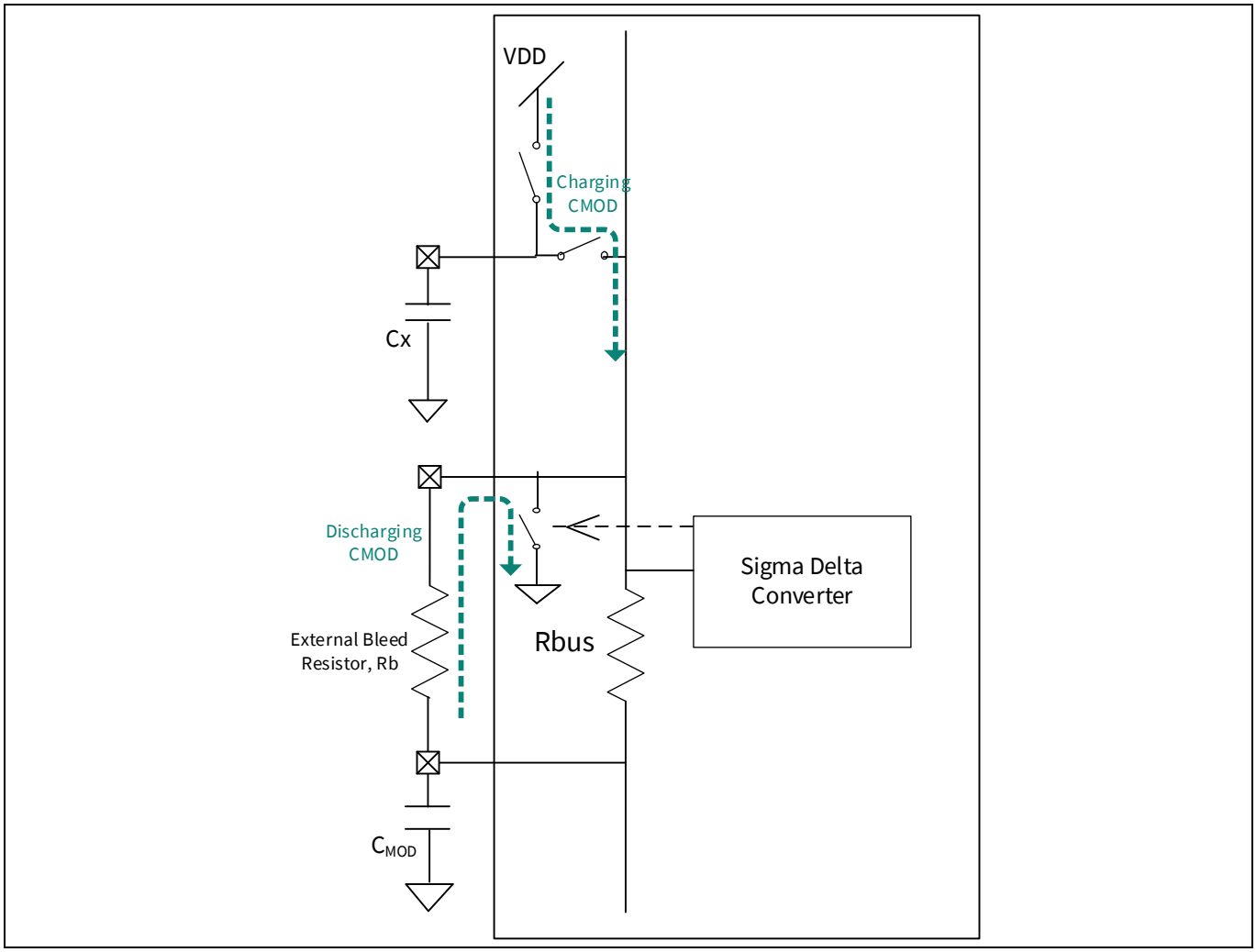

Figure 10 外部抵抗方式

Table 3 電流ソース方式の比較

	IDAC ソース	IDAC シンク	外部抵抗
DAC リソース	必要	必要	不要
外部ブリード抵抗	不要	不要	必要
電源ノイズ	影響されない	影響される	影響される
指伝導ノイズ	影響される VDAC で Vref を増加させることが必要	影響されにくい	影響されにくい
シールド電極	SIO ピン	任意の GPIO ピン	任意の GPIO ピン

3.2.4 チューニング

SmartSense 自動チューニングは、電源投入時に CAPSENSE™ ボタンを自動的にチューニングし、使用中も最適なボタン性能を維持する先進技術です。設計サイクルタイムを削減し、プロセスのばらつきに適応します。パラメーターをさらに制御することが必要な場合、または C_p が高い場合は、CAPSENSE™ パラメーターを手動でチューニングすることもできます。チューニング方法 では、チューニング方法の選択について説明します。CAPSENSE™ 性能のチューニング では、自動チューニングおよび手動チューニングのプロセスを説明します。

3.2.4.1 自動チューニング機能

設計サイクルタイムの短縮

Figure 11 に、SmartSense 自動チューニングがどのように設計サイクルタイムを著しく短縮するかを説明します。

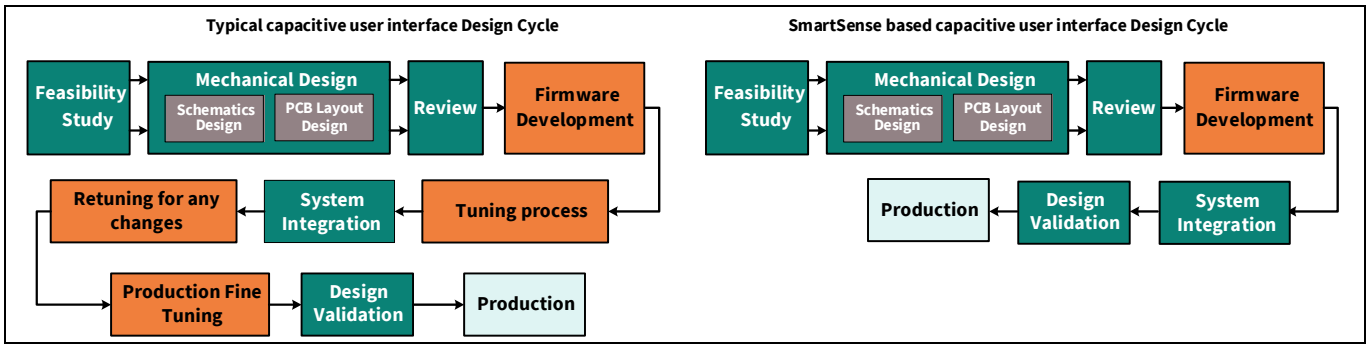

Figure 11 設計サイクルの比較

次の例は、SmartSense 自動チューニングがどのように設計サイクルタイムを短縮させ、プラットフォームの設計を可能にするかを説明します。Figure 12 に 21 インチのラップトップモデルのマルチメディアキーを示し、Figure 13 に 15 インチ モデルのマルチメディアキーを示します。キーは、同じ機能とサイズを持つ CAPSENSE™ ボタンです。しかし、モデル間でデザインを直接移植できません。理由は、21 インチ モデルは、ボタン間の間隔がより広く、ボタンと CAPSENSE™ コントローラー間の配線がより長いためです。デザインを再調整する必要があります。SmartSense 自動チューニングは、開発者が同じデザインを別のモデルへ移植する際にデザインサイクル時間を大きく削減することを可能にします。

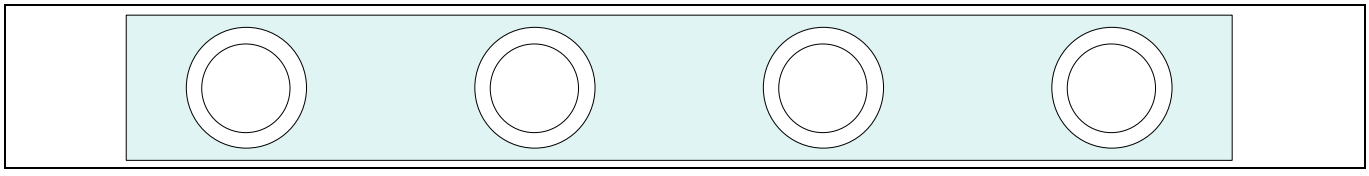

Figure 12 21 インチモデル用のラップトップ マルチメディアキー

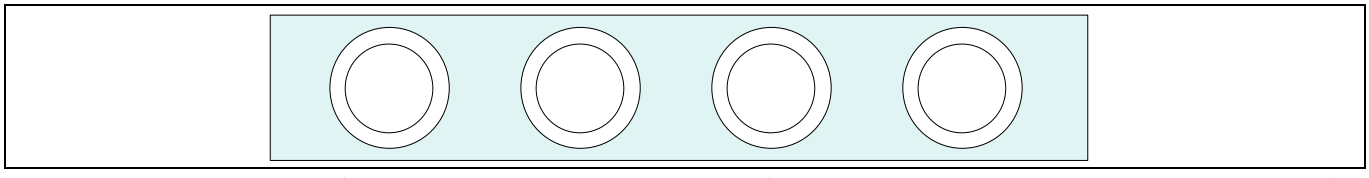

Figure 13 15 インチ モデル用のラップトップ マルチメディア キー (同一の機能およびボタン サイズ)

プロセスのばらつきに対する耐性

C_P は、プリント基板のレイアウトおよび配線の長さ、基板製造プロセスのばらつき、または複数購買サプライチェーン内のベンダー間のプリント基板のばらつきによって異なる可能性があります。ボタンの感度は C_P に依存します。C_P 値が高くなると感度が低下し、指を触れる信号の振幅が減少します。C_P の変化により、ボタンが高感度になりすぎたり、感度が十分でなくなったり、または動作しなくなってしまうことがあります。そのとき、ユーザーはシステムを再調整し、場合によってはユーザーインターフェースのサブシステムを再検証する必要があります。SmartSense 自動チューニングはこのような問題を解決します。

使いやすさ

SmartSense 自動チューニングでは、すべてのパラメーターに関する深い知識を持っていなくても CAPSENSE™ アプリケーションを迅速で容易に設計できます。

3.2.4.2 手動チューニング機能

詳細制御

手動チューニングは、各 CAPSENSE™ パラメーターを選択することができます。これは、ノイズが高いシステムなどの特別な動作条件のシステムに重要です。ノイズの多いシステムで自動チューニングの使用は、誤ったボタン タッチの原因になる可能性があります。手動で指閾値を増加させることで、システムのノイズ耐性を向上させます。

高寄生容量

自動チューニングは、5～45 pF の範囲の C_P 値に対応するように設計されています。長い配線や大きなボタンサイズのため C_P 値が 45 pF を超える場合は、手動チューニングを使用してください。

不良検出アルゴリズム

自動チューニングでは、起動および実行中でパラメーターを変更することが可能です。これにより、不良検出アルゴリズムの作成が難しくなることがあります。

ベースラインカウントが工場出荷時にフラッシュに格納された値の範囲内であるかどうかを確認する不良検出アルゴリズムを考えてみましょう。格納された値が 12 ビットのスキャン分解能で測定された 3000 カウントであると仮定します。ベースライン カウントは C_P の平均カウントであり、スキャン分解能は静電容量の計測の分解能です (これらの用語の詳細な説明については、raw カウント および スキャン分解能 を参照してください)。自動チューニングを使用した場合、物理的条件または環境条件が変化するためスキャン分解能パラメーターが 13 ビットに設定されることがあります。スキャン分解能が 13 ビットに設定されると、ベースラインは 6000 カウントになり、システムは不調になります。

RAM とフラッシュの使用の削減

自動チューニング アルゴリズムには、より大量のフラッシュと RAM が必要です。Table 4 に、4 つのセンサーを使用する CAPSENSE™ デザインに必要なメモリを示します。

Table 4 手動チューニングと自動チューニングの必要なメモリの比較

チューニング方法	PSOC™ 3		PSOC™ 5LP	
	RAM (バイト)	フラッシュ (バイト)	RAM (バイト)	フラッシュ (バイト)
手動チューニング	207	6345	384	5104
自動チューニング	292	8282	488	6152

3.2.4.3 ノンブロッキングアーキテクチャ

CAPSENSE™ スキャンには 3 つの段階があります。

1. **前スキャン**: ハードウェアの準備
2. **ハードウェアスキャン**: 実際のハードウェアスキャン
3. **後スキャン**: 結果の処理と保存

前スキャンと後スキャンのみは、CPU がコードを実行する必要があります。コードアーキテクチャは、ハードウェアスキャンが完了するまで待ちません。代わりに CPU はコードの次のラインを実行し、ハードウェアはスキャンが完了し、割込みを生成すると、ISR で後スキャンコードを実行します。このノンブロッキングアーキテクチャは、CPU が複数のアプリケーションを処理するデザインに役立ちます。

3.3 CAPSENSE™ PLUS

CAPSENSE™ PLUS は、CAPSENSE™ に加えて機能を実行する PSOC™ デバイスを表します。PSOC™ 3 および PSOC™ 5LP デバイスはこのカテゴリに該当します。Figure 14 に示すように、これらのデバイスが提供する様々な機能により、多数のシステム機能を単一のチップに組み込めます。これにより、基板サイズ、BOM コスト、および消費電力を削減できます。

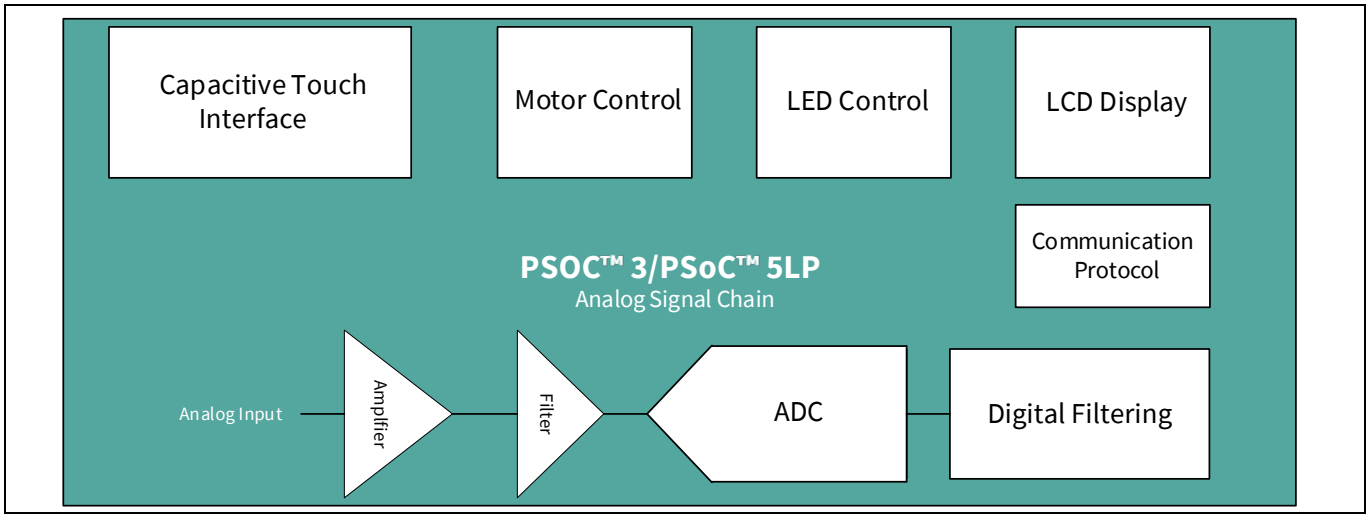

Figure 14 システム内の CAPSENSE™ PLUS のその他の機能

当社の Web サイトのアプリケーションにアクセスして、CAPSENSE™ Plus と PSOC™ 3 および PSOC™ 5LP デバイスの組み合わせが付加価値をもたらす領域をご覧ください。

- ヘルスケアとライフスタイル
- 民生用電子機器
- スマートホームとビルディング
- 家電製品

4 CAPSENSE™ 設計ツール

インフィニオンは、CAPSENSE™ 静電容量タッチセンシングアプリケーションの開発用に幅広いハードウェアおよびソフトウェアツールを提供しています。注文情報は、参考資料 のデータシートを参照してください。

4.1 PSOC™ Creator

インフィニオンの PSOC™ Creator は、統合開発環境を提供します。PSOC™ Creator は、ハードウェア コンフィギュレーションとソフトウェア開発を単一の統合ツールに組み合わせています。アプリケーションは、コンポーネントライブラリを使用するドラッグアンドドロップ方式の設計環境で開発されます。

PSOC™ Creator には CapSense_CSD コンポーネントがあります。このコンポーネントは、スイッチトコンデンサ回路、アナログバス、コンパレータ、デジタルカウント機能、および高レベルソフトウェアルーチン (API) を使用して静電容量タッチセンサーを実装します。I²C、SPI、UART、タイマ、PWM、アンプ、ADC、LCD などの他の機能を実装するために使用できるその他アナログおよびデジタルコンポーネントがあります。Figure 15 に、コンポーネントカタログからドラッグし、TopDesign に配置した CapSense_CSD コンポーネントを示します。

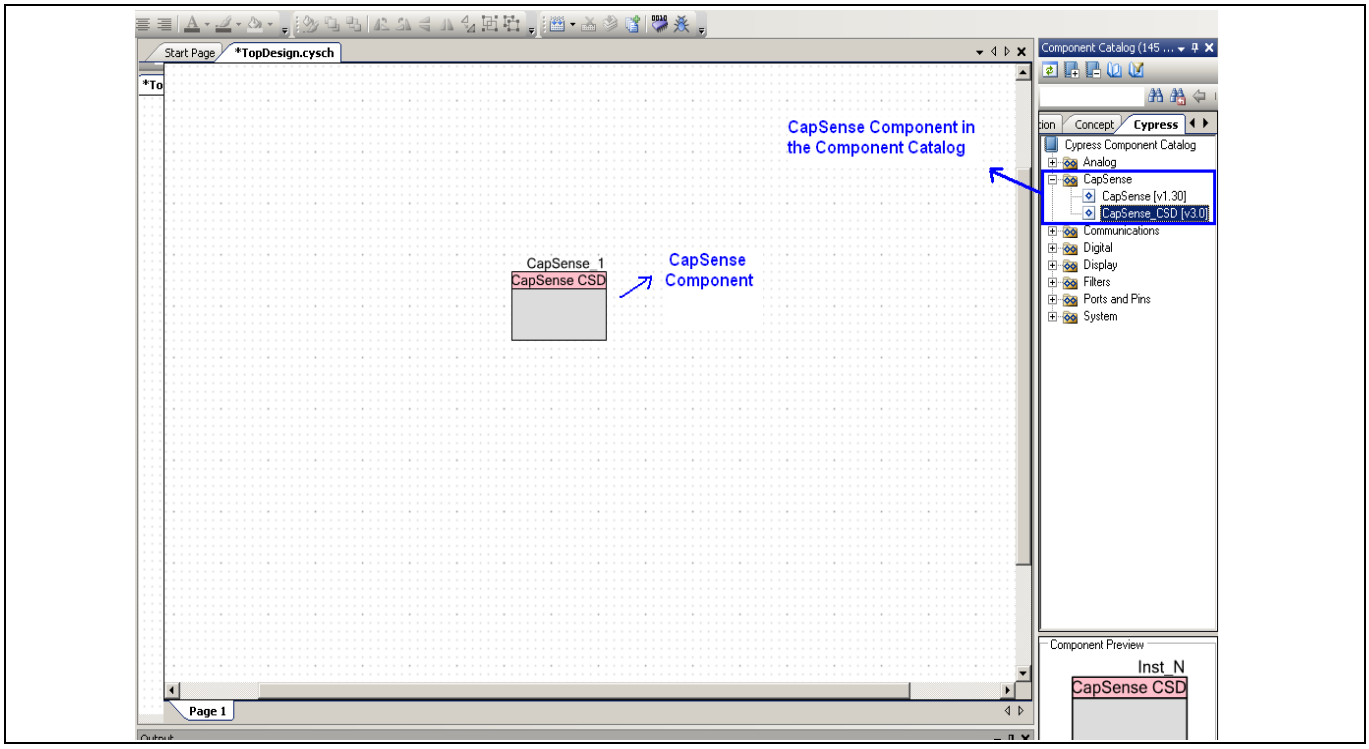

Figure 15 PSOC™ Creator の TopDesign

4.2 ハードウェアキット

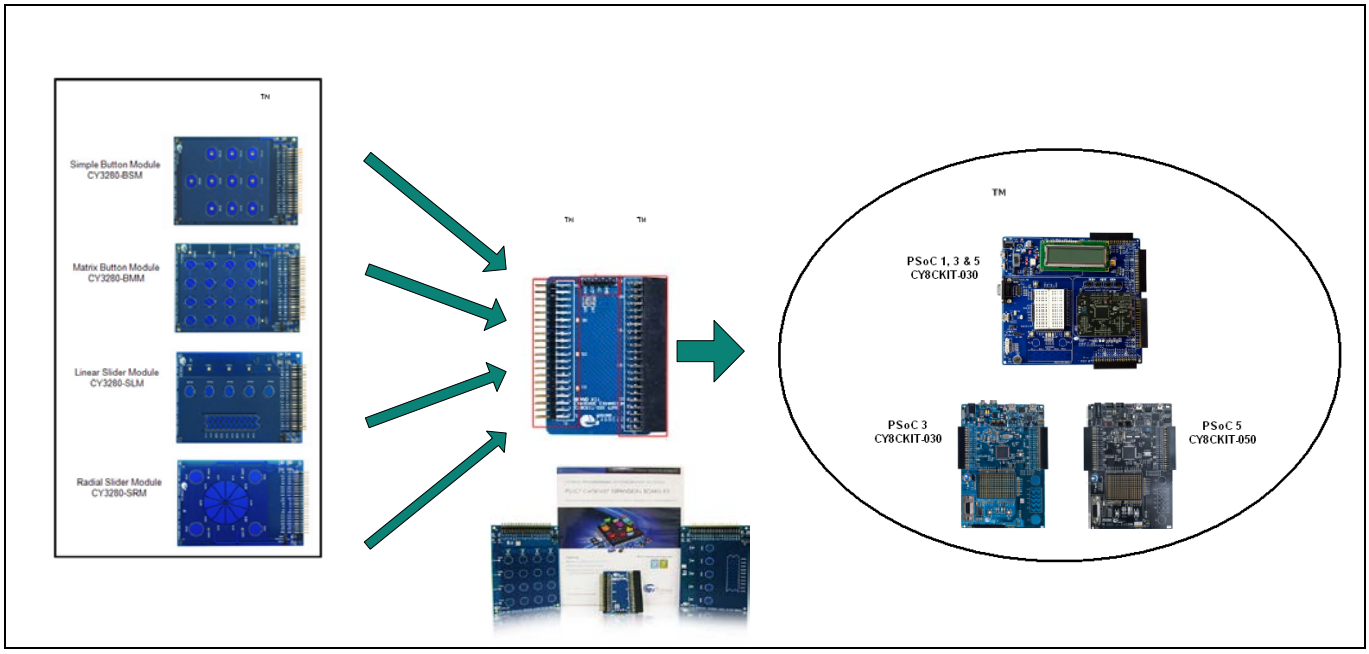

Figure 16 CAPSENSE™ ハードウェア

Note: インフィニオンは、ここで説明する Universal CAPSENSE™ モジュール ボードおよび PSOC™ CAPSENSE 拡張ボード キットの販売を終了しました。このハードウェアをすでに所有している可能性のある顧客のために、コンテキスト情報として残しておきます。

4.2.1 PSOC™ 3 および PSOC™ 5LP 開発キット

PSOC™ 3 および PSOC™ 5LP 開発キットは CAPSENSE™ 機能に対応しています。

- CY8CKIT-030 PSOC™ 3 開発キット
- CY8CKIT-050B PSOC™ 5LP 開発キット

4.2.2 ユニバーサル CAPSENSE™ モジュール基板

インフィニオンは、ニーズを満たす各種の CAPSENSE™ センサー、LED およびインターフェースを備えたユニバーサル CAPSENSE™ モジュール基板を提供しています。

- CY3280-BSM シンプルボタン モジュール
- CY3280-BMM マトリックス ボタン モジュール
- CY3280-SLM リニア スライダー モジュール
- CY3280-SRM ラジアル スライダー モジュール
- CY3280-BBM ユニバーサル CAPSENSE™ プロトタイプ モジュール

4.2.3 PSOC™ CAPSENSE™ 拡張基板キット

CY8CKIT-031 PSOC™ CAPSENSE™ の拡張基板キットは、任意の PSOC™ 3 と PSOC™ 5LP 開発キットを任意のユニバーサル CAPSENSE™ モジュール基板に接続します。CY8CKIT-031 は、インターフェース基板および 2 個のモジュール基板 (CY3280-SLM と CY3280-BMM) を提供します。

5 CapSense_CSD コンポーネント

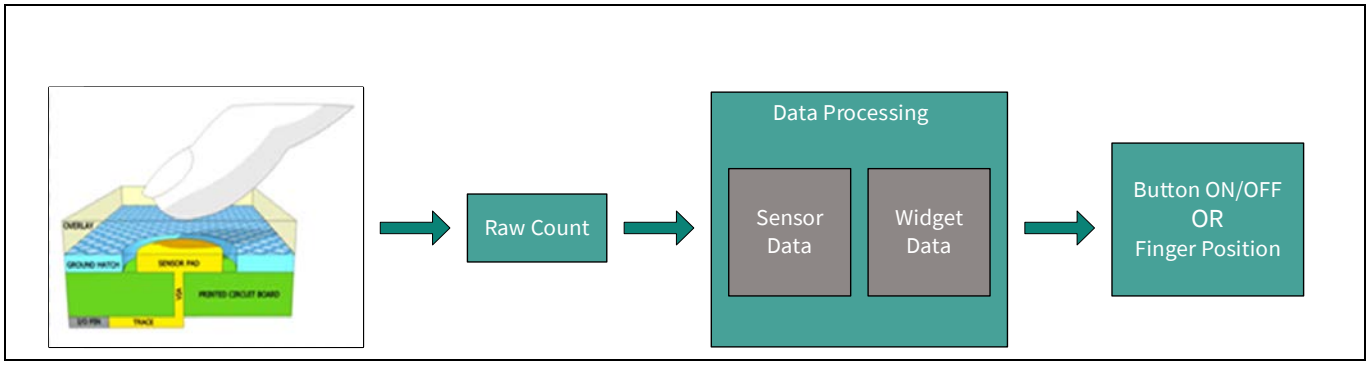

Figure 17 CapSense_CSD コンポーネント ブロック図

CapSense_CSD コンポーネントは、完全な CAPSENSE™ システムを提供します。コンポーネントには、高レベルと低レベルに分類された、いくつかのパラメーターがあります。パラメーターは、グローバル アレイを使用してファームウェア内で相互に通信します。

高レベルパラメーターは、CAPSENSE™ システムが raw カウント値を、センサーのオン/オフ状態などの便利な情報に変換する処理方法を制御します。高レベルパラメーターの例は、指閾値、ノイズ閾値、デバウンスカウントなどです。高レベル パラメーター を参照してください。

低レベルパラメーターは、物理層での CAPSENSE™ システムの動作を制御します。物理層で静電容量は raw カウントに変換されます。低レベルパラメーターの例は、IDAC 範囲、IDAC 値、スキャンクロックなどです。低レベルパラメーター を参照してください。

CapSense_CSD コンポーネントは、ウィジェットと呼ばれるタッチインターフェースの各種を提供します。ウィジェットは 1 つ以上のセンサーで構成されます。これらは、スライダーやタッチ パッドなどのインターフェースオブジェクトを表します。ウィジェットのタイプは、ボタン、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、近接センサーを含んでいます。Figure 18 にスライダーの例を示します。スライダーウィジェットを形成するには、7 つのセンサーを狭い間隔で直線上に配置します。

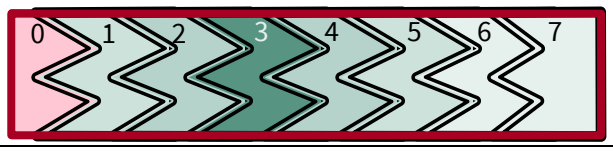

Figure 18 7 セグメント スライダー

5.1 パラメーターのまとめ

Figure 19 に、PSOC™ Creator の CapSense_CSD コンポーネントのスクリーンショットおよびコンフィギュレーションウィンドウを示します。また、コンポーネントをダブルクリックするか、右クリックして「Configure」を選択することでコンフィギュレーションウィンドウを開きます。

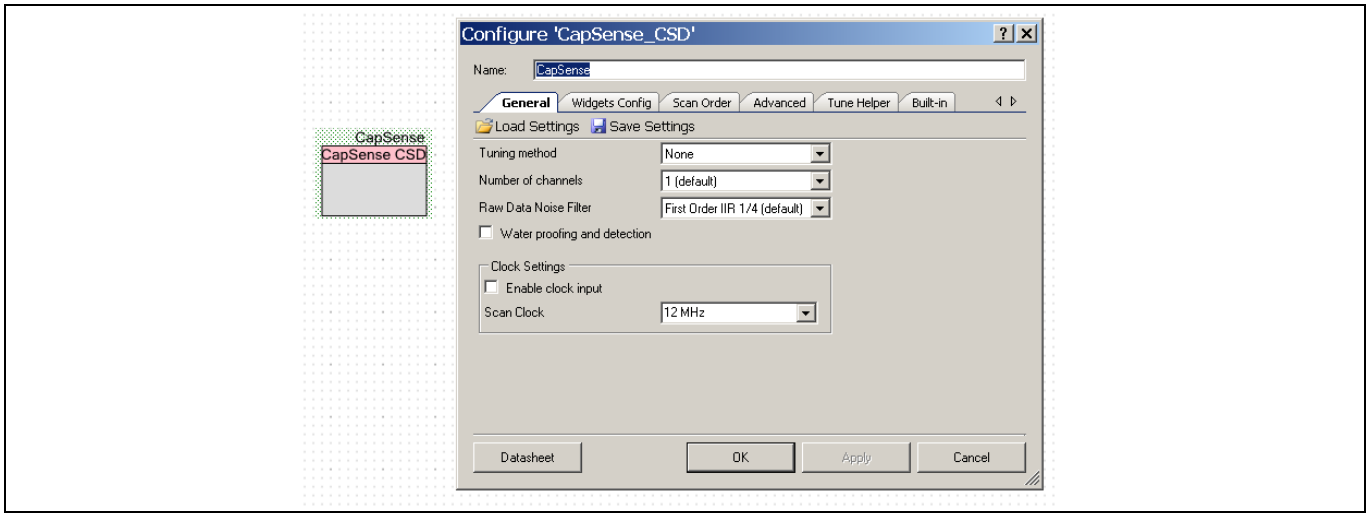

Figure 19 PSOC™ Creator の CAPSENSE™ コンポーネント

コンフィギュレーションウィンドウには、いくつかのパラメーターが異なるタブに配置されています。Table 5 に、このウィンドウにあるパラメーターをまとめ、関連するリンクを提供します。

Table 5 CapSense_CSD コンポーネントのコンフィギュレーションウィンドウパラメーター

全般	ウィジェット設定	スキャン順序	詳細設定		チューナーヘルパー
Tuning method (チューニング方法)	Add widget (ウィジェット追加)	Analog switch divider (アナログスイッチ分周器)	Analog switch drive source (アナログスイッチドライブソース)	Shield (シールド)	Tune Helper (チューナーヘルパー) のイネーブル
Number of channels (チャネル数)	Finger threshold (指閾値)	IDAC value (IDAC 値)	Multiple analog switch Divider (複数のアナログスイッチ分周器)	Inactive sensor connection (非アクティブなセンサー接続)	EZI2C コンポーネントのインスタンス名
Raw data noise filter (raw データノイズフィルタ)	Noise threshold (ノイズ閾値)	Move to channel 1/move to channel 0 (チャネル 1/チャネル 0 に変更)	Analog switch divider (アナログスイッチ分周器)	Guard sensor (ガードセンサー)	
Water proofing and detection (耐水性および検知)	Hysteresis (ヒステリシス)	Sensitivity (感度)	Scan speed (スキャン速度)	Current source (電流ソース)	
Enable clock input (クロック入力のイネーブル)	Debounce (デバウンス)		PRS EMI reduction (PRS EMI 減少)	IDAC range (IDAC 範囲)	

5 CapSense_CSD コンポーネント

全般	ウィジェット設定	スキャン順序	詳細設定		チューナーヘルパー
Scan clock (スキャンクロック)	Scan resolution (スキャン分解能)		Sensor auto reset (センサー自動リセット)	Number of bleed resistors, channel 0/ channel 1 (ブリード抵抗, チャネル 0/チャネル 1 の数)	
	Number of sensor elements (センサーエレメント数)		Widget resolution (ウィジェット分解能)	Digital resource implementation, channel 0/ channel 1 (デジタルリソース実行, チャネル 0/チャネル 1)	
	API resolution (API 分解能)		Negative noise threshold (負のノイズ閾値)	Voltage reference source (電圧基準ソース)	
	Position noise filter (位置ノイズフィルタ)		Low baseline reset (低ベースラインリセット)		
	No of dedicated sensor elements (専用センサーエレメント数)				

低レベル パラメーター
高レベル パラメーター
ウィジェットパラメーター
その他のシステムパラメーター

5.2 グローバルアレイ

CAPSENSE™ システムは、環境の変化がある場合の適切な検出と動作を確保するために、いくつかのグローバルアレイを使用します。これらのアレイは手動で変更できませんが、デバッグ用に検査できます。

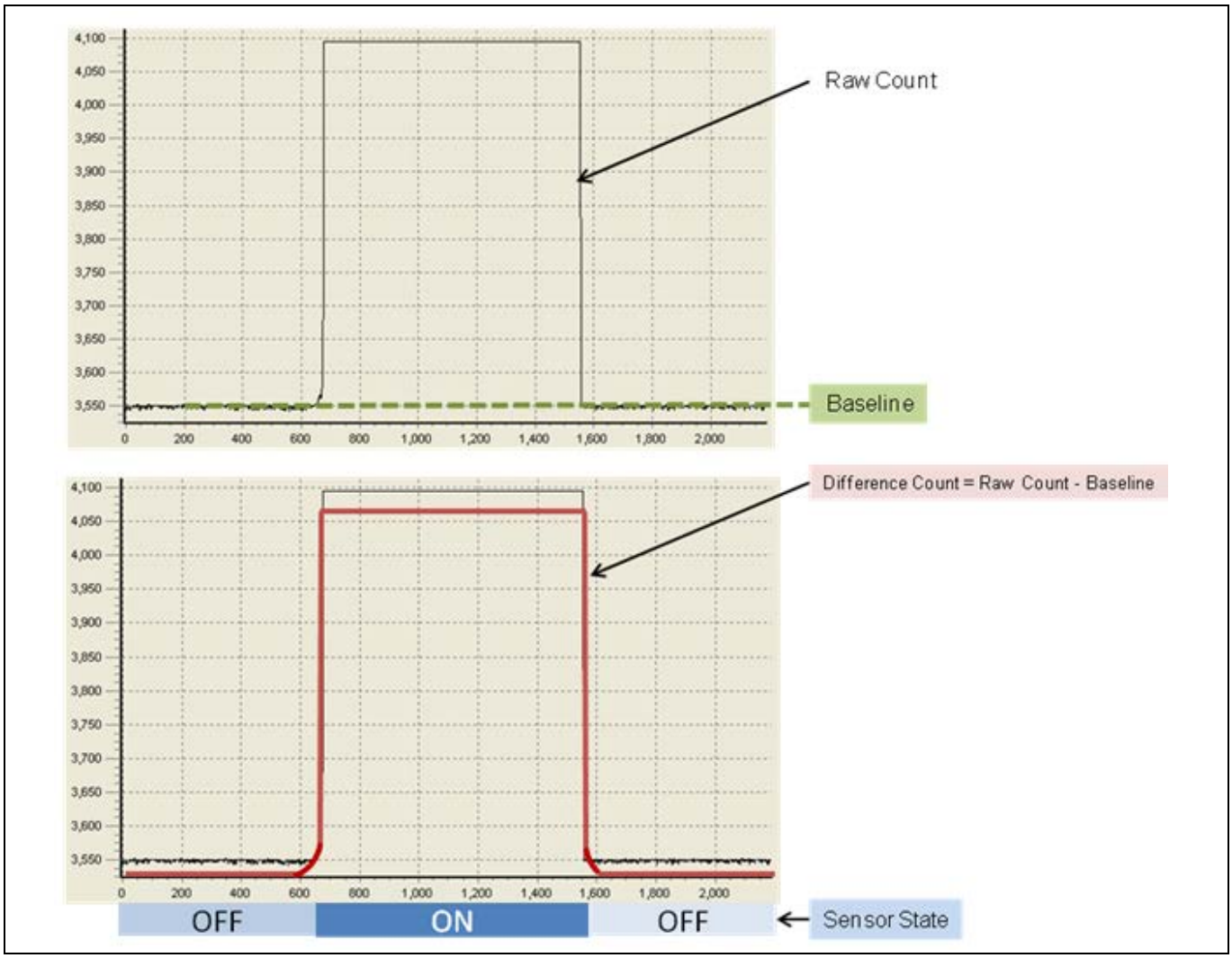

Figure 20 グローバルパラメーター

5.2.1 raw カウント

CAPSENSE™ コントローラーハードウェアは静電容量を計測し、raw カウントと呼ばれるデジタル形式で結果を提供します。raw カウントの値は、センサーの静電容量が増加するとともに増加します。

5.2.2 ベースライン

センサーの raw カウント値は、温度や湿度などの環境変化により、徐々に変わります。このような徐々に現われる変動はベースライン値によって補正されます。ベースラインは、ソフトウェアアルゴリズムを使用して、raw カウントの徐々に起こる変化を追跡します。これは raw カウントの突然の変化に影響されにくいローパスフィルタです。ベースライン値は、差分カウントを計算するための基準レベルを提供します。

5.2.3 差分カウント

差分カウントとは、センサーの raw カウントとベースラインの差です。通常は、センサーが非アクティブの場合、差分カウントはゼロです。センサーにタッチすると、raw カウントが増加するため、正の差分カウント値が発生します。

5.2.4 センサー状態

各センサーの状態は、ボタンがオンの場合は 1 を表し、ボタンがオフの場合は 0 として表します。すべてのセンサーのオン/オフ状態はバイトアレイに格納されます。各アレイエレメントは、8 個のセンサーのオン/オフ状態を保持できます。

5.3 高レベル パラメーター

高レベルパラメーターでは、センサーのオン/オフ状態やスライダー上の指の位置の推定値などの情報を生成するために raw カウントを処理する方法を定義します。Figure 21 および Equation 4 に、高レベルパラメーターの概要を示します。

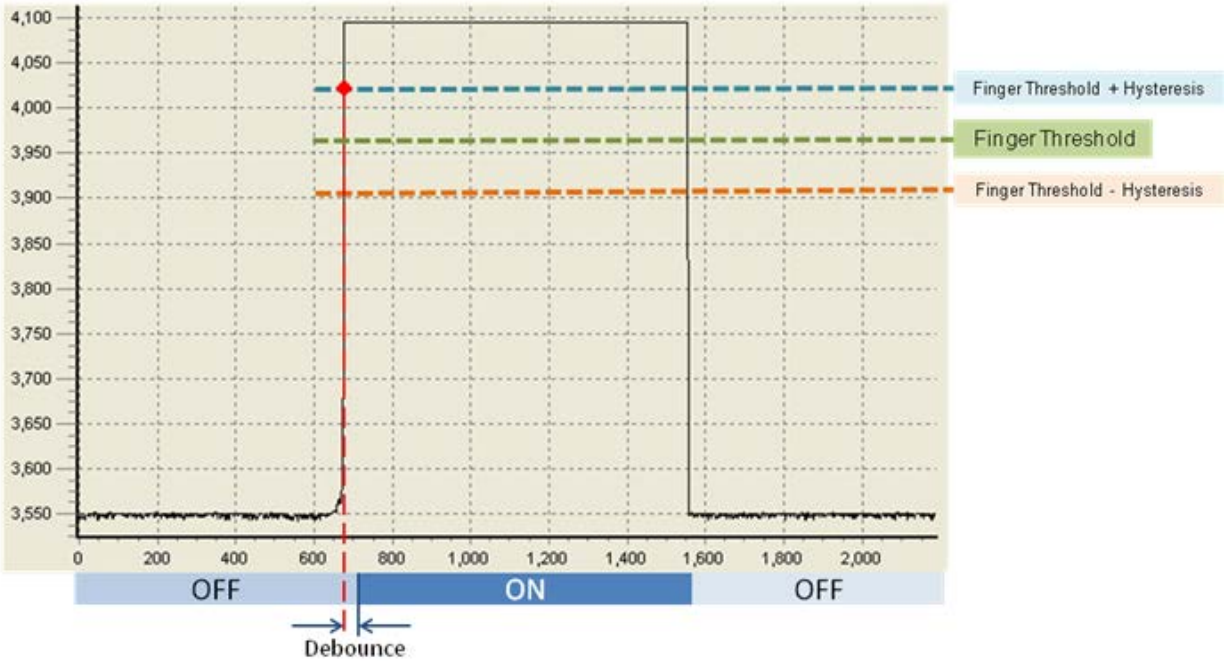

Figure 21 高レベル パラメーター

$$if\ (Difference\ Count \geq Finger\ Threshold\ + Hysteresis)\ \&\ (Sample\ Count\ \geq Debounce),\ Sensor\ State = ON$$

$$if\ (Difference\ Count\ \leq Finger\ Threshold - Hysteresis),\ \ Sensor\ State = OFF$$

Equation 4

ここで、

サンプル カウント = 指閾値+ヒステリシスより大きい測定値のサンプル数

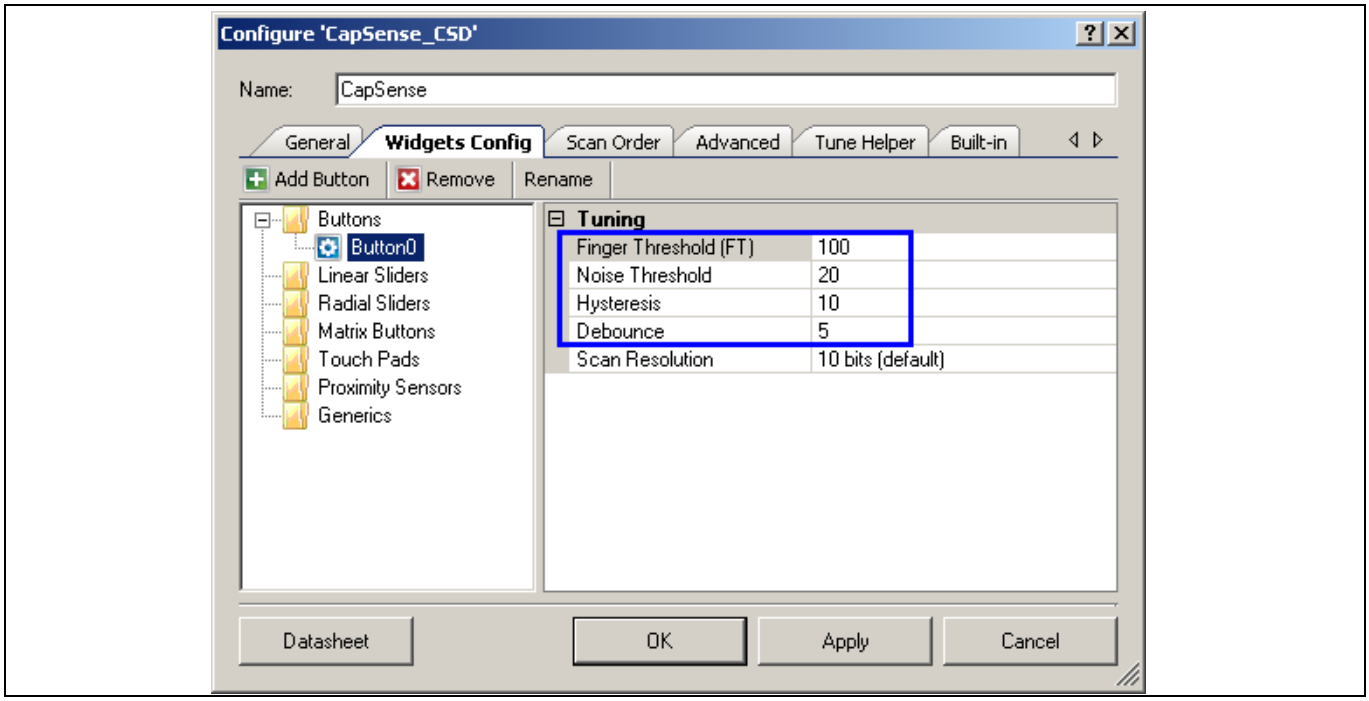

Figure 22　高レベルパラメーター

5.3.1　指閾値

指閾値パラメーターは、指のタッチに対するセンサーの感度を定義します。これは Equation 4 で定義されるヒステリシスパラメーターと併用してセンサーの状態を判断します。値の範囲は 0～255 です。

5.3.2　ヒステリシス

ヒステリシスパラメーターは Equation 4 で定義されたように、指閾値と併用してセンサーの状態を判断します。ヒステリシスはノイズの多い遷移の耐性を向上させます。これはボタンのデバウンス機能です。タッチ状態は、差分カウントが指閾値より少し大きくなるまで、オフのままになります。タッチ状態は、差分カウントがノイズ閾値より少し小さくなるまで、オンのままになります。これにより、差分カウントがノイズを含んでおり、ノイズと指閾値の間の中間にある場合、タッチ/非タッチのステートマシンがトリガーすることを防ぎます。

値の範囲は 0～255 です。ヒステリシスパラメーターは、指閾値パラメーターより低く設定する必要があります。

5.3.3　デバウンス

デバウンスパラメーターは、センサーのオフからオンへの遷移にカウンターを加えます。センサーがオフからオンになるには、特定のサンプル数の間、差分カウントは (指閾値 + ヒステリシスレベル) より大きい値を維持する必要があります。

値の範囲は 1～255 です。1 をセットするとデバウンスはありません。

5.3.4 ノイズ閾値

個々のセンサーに対して、ノイズ閾値パラメーターは raw カウントの上限を設定してベースライン値を更新します。スライダーセンサーに対しては、raw カウントの下限を設定して重心計算で結果をカウントします。

値の範囲は 3～255 です。ユーザーモジュールが正常に動作するためには、ノイズ閾値を (指閾値 - ヒステリシス値) より高く設定しないでください。

5.3.5 負のノイズ閾値および低ベースラインリセット

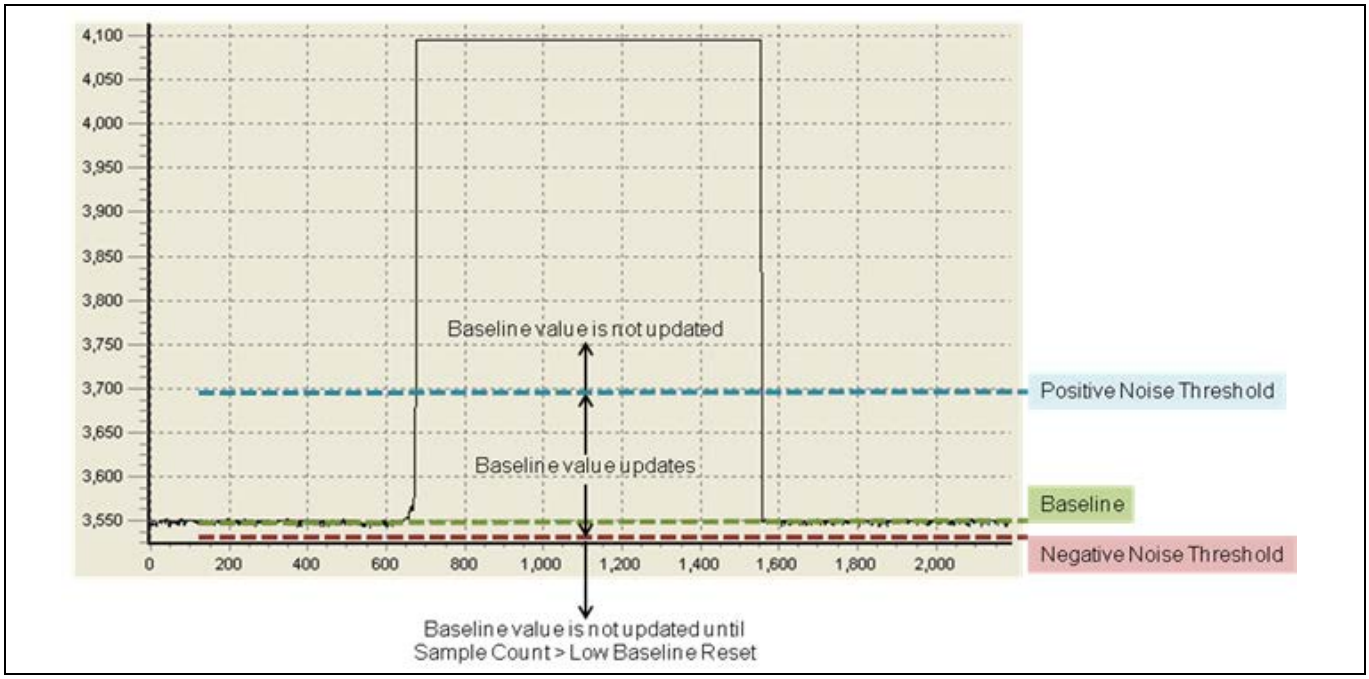

Figure 23 ノイズ閾値とベースライン更新パラメーター

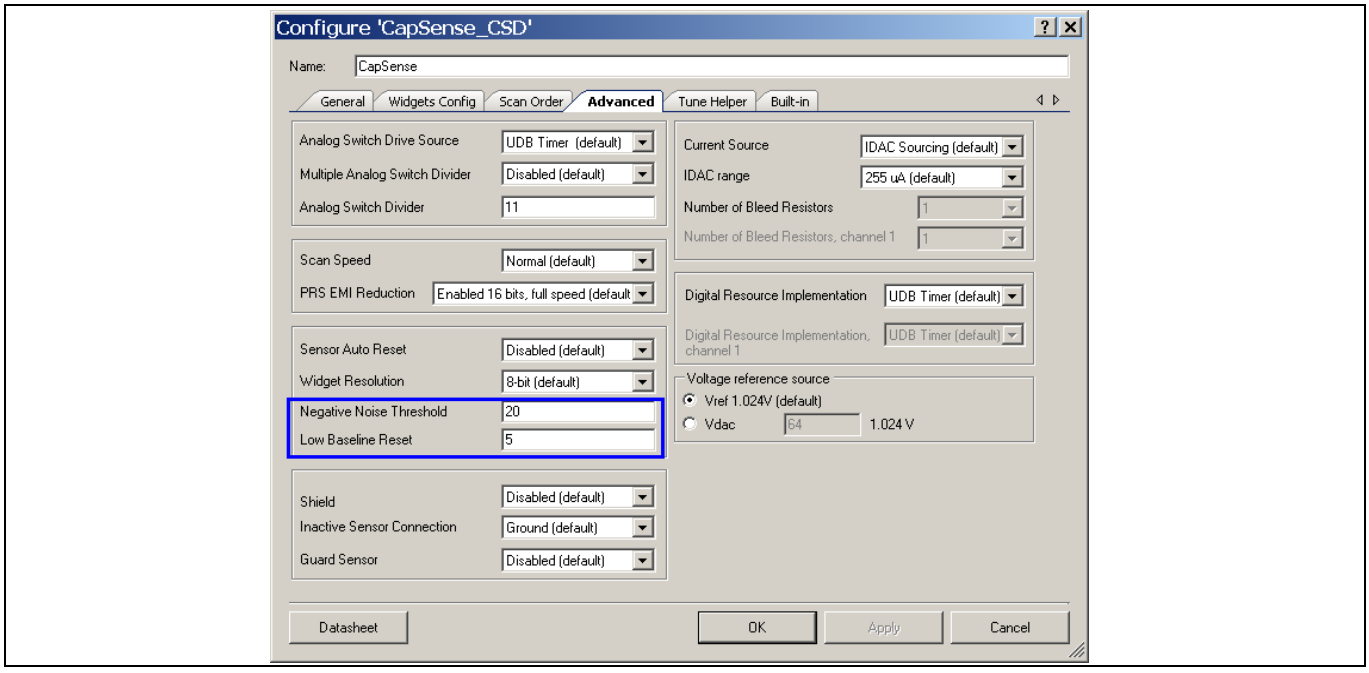

Figure 24 負のノイズ閾値および低ベースラインリセットパラメーター

負のノイズ閾値パラメーターは、負の差分カウント閾値として使用されます。raw カウントが、低ベースラインリセットパラメーターによって指定されたサンプル数の間、(ベースライン - 負のノイズ閾値) より小さい場合、ベースラインは新しい raw カウント値に設定されます。値の範囲は 0〜255 です。

低ベースラインリセットパラメーターは、負のノイズ閾値パラメーターと併用されます。これは、ベースライン値のリセットが必要となる異常に低いサンプルの数をカウントするものです。起動時に指が置かれている状態の補正をするために使用されます。値の範囲は 0〜255 です。

5.3.6 センサー自動リセット

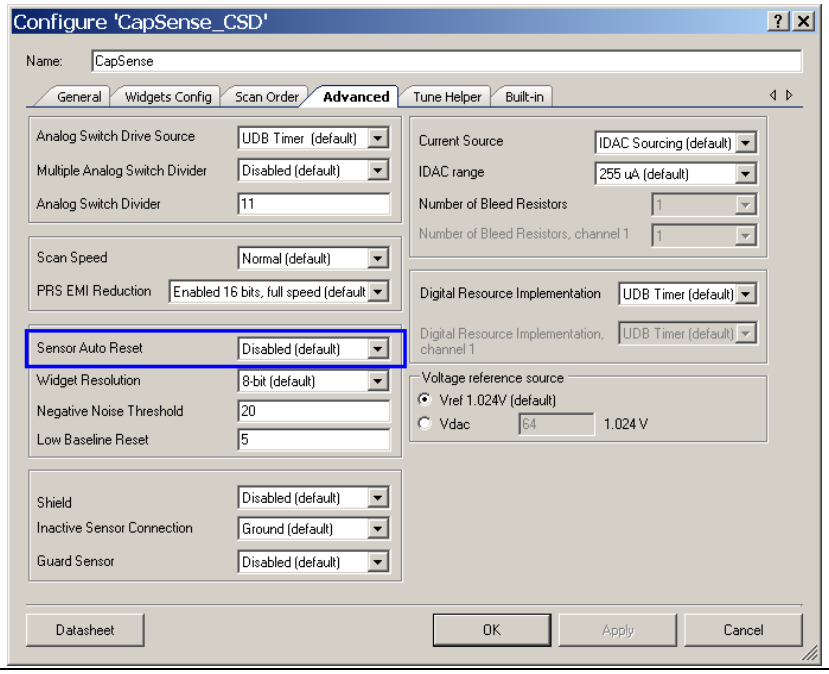

Figure 25 センサー自動リセットパラメーター

このパラメーターは、ベースライン値が常に更新されるか、差分カウントがノイズ閾値より低い場合にのみ更新されるかを指定します。

Enabled (有効): ベースラインは常に更新されます。これは、センサーの最大時間 (標準値は 5〜10 秒) を制限しますが、センサーに触れずに raw カウントが誤って上がった場合に、センサーが永久的にオンになることを防ぎます。この突然の上昇の原因には、大幅な電源電圧の変化や高エネルギー RF ノイズソース、急激な温度変化などがあります。

Disabled (無効): ベースライン値は差分カウントがノイズ閾値パラメーターを下回った場合にのみ更新されます。

5.3.7 ウィジェット分解能

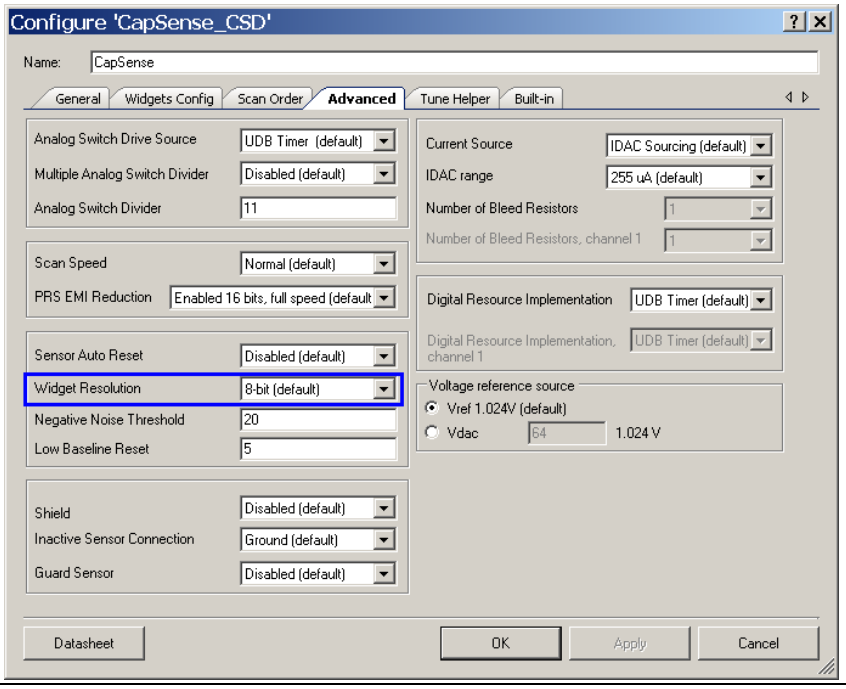

Figure 26 ウィジェット分解能パラメーター

このパラメーターは、信号カウントに 8 ビット変数か 16 ビット変数を使用するかを決めます。信号カウントは Figure 21 に示すように、raw カウントとベースラインの差です。このパラメーターはウィジェットレベルで設定されます。ウィジェットのすべてのセンサーは同じ分解能を持ちます。

ほとんどの場合、適切な初期設定値が 8 ビットです。ただし、薄いオーバーレイまたは大きいセンサー領域による高信号レベルの設計では、16 ビットの分解能を選択する必要があります。スライダーが隣接センサーの相対信号レベルを用いて指の位置を計算し、信号が飽和して円滑でない動作を引き起こすため、16 ビットの分解能は必要になることがあります。

5.3.8 フィルタの選択

5.3.8.1 raw データノイズフィルタ

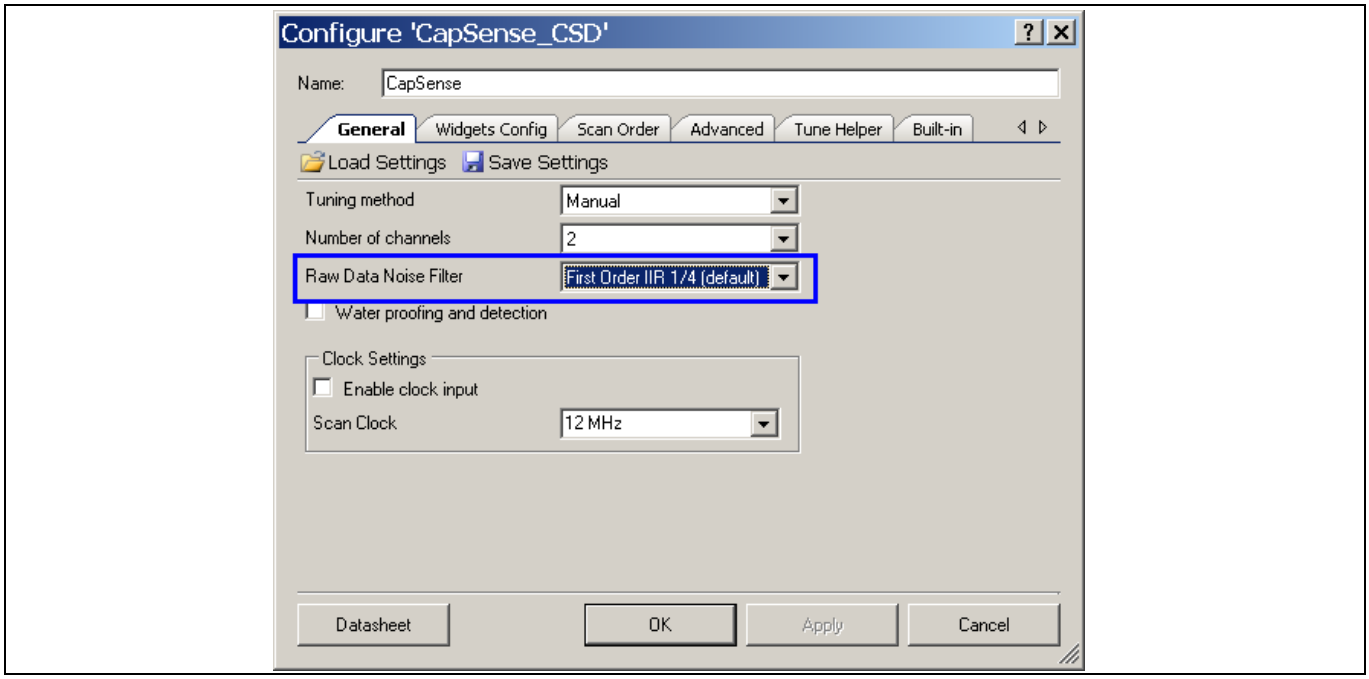

Figure 27 raw データノイズフィルタパラメーター

raw カウントは CAPSENSE™ システムの raw 出力であり、固有のノイズがあります。raw カウントのノイズは、ソフトウェアフィルタを使って最小限にできます。raw データノイズフィルタパラメーターは、いくつかのソフトウェアフィルタを選んで raw カウントおよび指の位置データを整理できます。各フィルタタイプの使い方は、ソフトウェアのフィルタリング を参照してください。

5.3.8.2 位置ノイズフィルタ

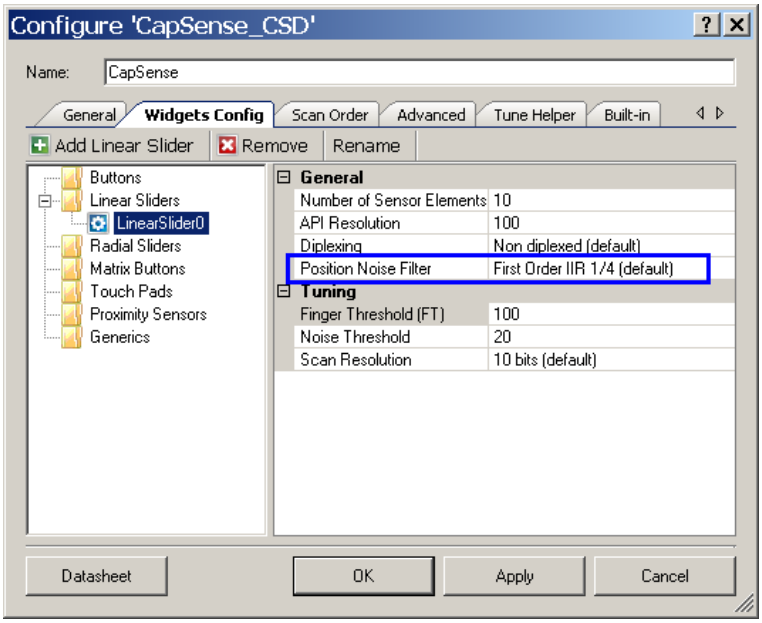

Figure 28 位置ノイズフィルタパラメーター

リニアスライダーやラジアルスライダーなどの一部のウィジェットタイプには、指位置出力があります。位置ノイズフィルタパラメーターは、いくつかのソフトウェアフィルタを選んで指位置出力を整理できます。

5.3.9 高レベルパラメーターの推奨事項

次の推奨事項は最適なパラメーター設定を選択するための出発点となります。

- **指閾値**: センサーがオンの状態で raw カウントの 75%に設定
- **ノイズ閾値**: センサーがオフの状態で raw カウントの 40%に設定
- **負のノイズ閾値**: ノイズ閾値に等しく設定
- **ヒステリシス**: センサーがオンの状態で raw カウントの 15%に設定
- **低ベースラインリセット**: 10 に設定
- **センサー自動リセット**: 設計要件に応じて設定
- **デバウンス**: 設計要件に応じて設定
- **ウィジェット分解能**: スライダーを使用し、薄いオーバーレイまたは大きいセンサー領域により信号が飽和する場合は 16 ビットを選択し、それ以外の場合は 8 ビットを選択
- **フィルタ選択**: 適切なフィルタを選択するために、ソフトウェアフィルタリング を参照してください。

5.4 低レベルパラメーター

低レベルパラメーターは、物理層で CAPSENSE™ システムの動作を定義し、静電容量から raw カウントへの変換に関連しています。

5.4.1 クロック設定

5.4.1.1 スキャンクロック

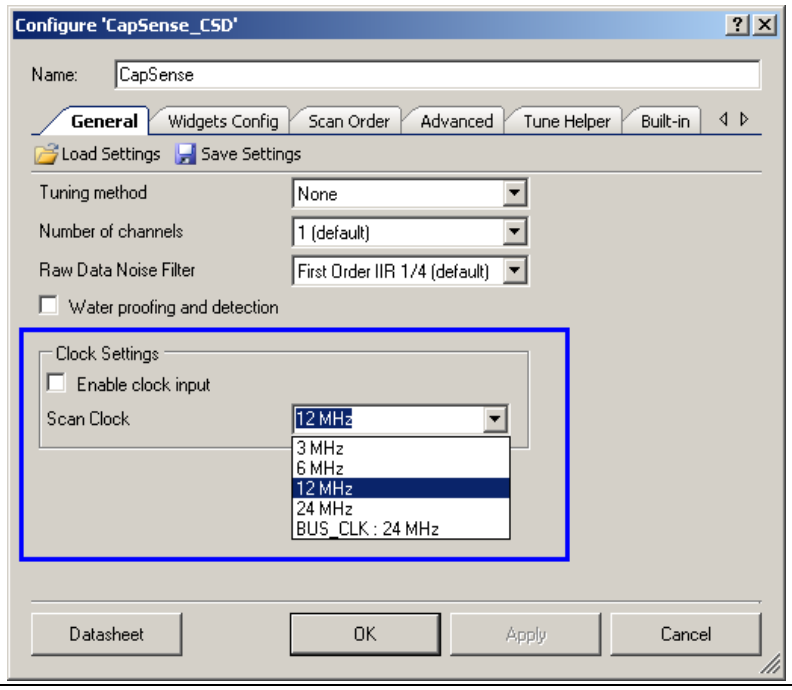

Figure 29 スキャンクロックパラメーター

スキャンクロックパラメーターは、CAPSENSE™_CSD ブロックのクロックソースを選択します。スキャンクロックは CPU 周波数に依存しません。これはマスタークロックから派生するため、マスタークロックの周波数はスキャンクロックパラメーター以上にする必要があります。

5.4.1.2 クロック入力のイネーブル

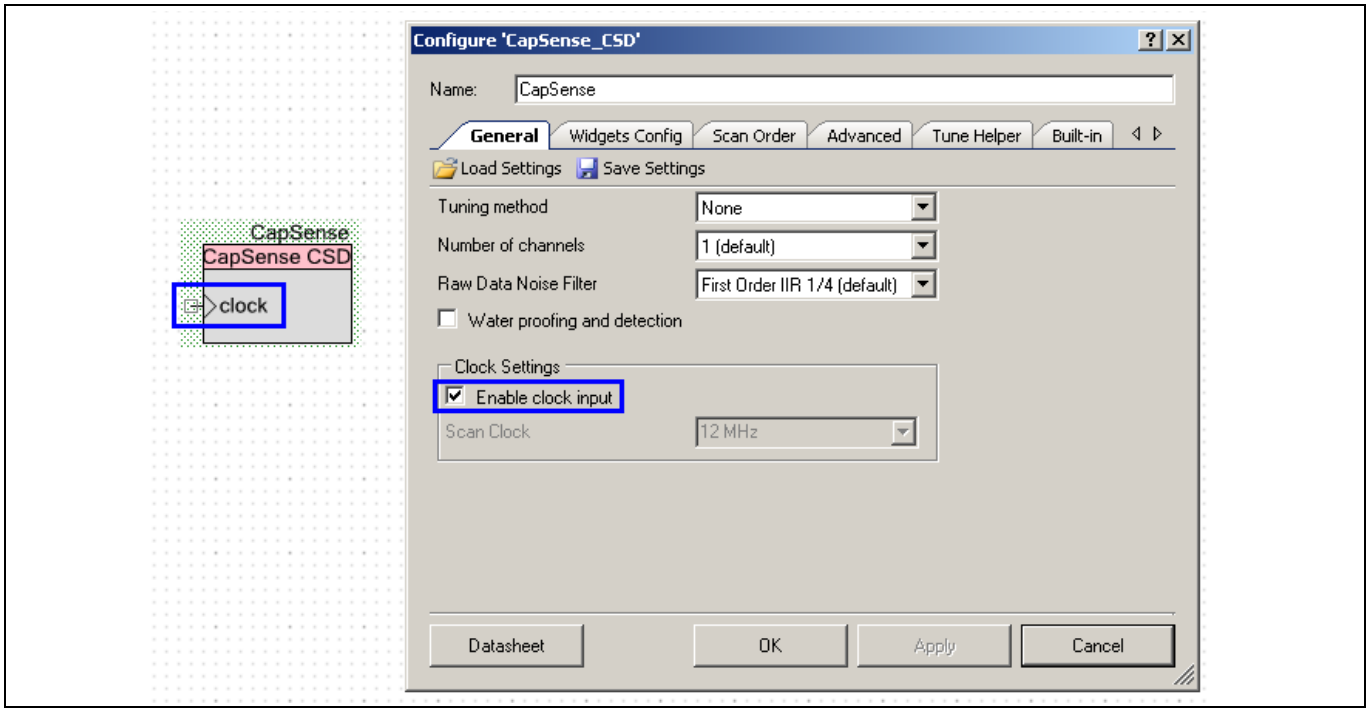

Figure 30 外部クロック入力パラメーター

クロック入力のイネーブル パラメーターを選択することは、クロックソースが CAPSENSE™_CSD ブロックの外部から供給されることを意味します。このオプションを選択した場合、Figure 30 に示すように、スキャンクロックの周波数は無効になり、端子が CapSense_CSD コンポーネントに現れます。システム内のデジタル信号および GPIO ピン経由で送られる外部信号は、この端子に接続できます。

5.4.2 アナログスイッチ分周器

5.4.2.1 アナログスイッチ分周器 (プリスケーラー)

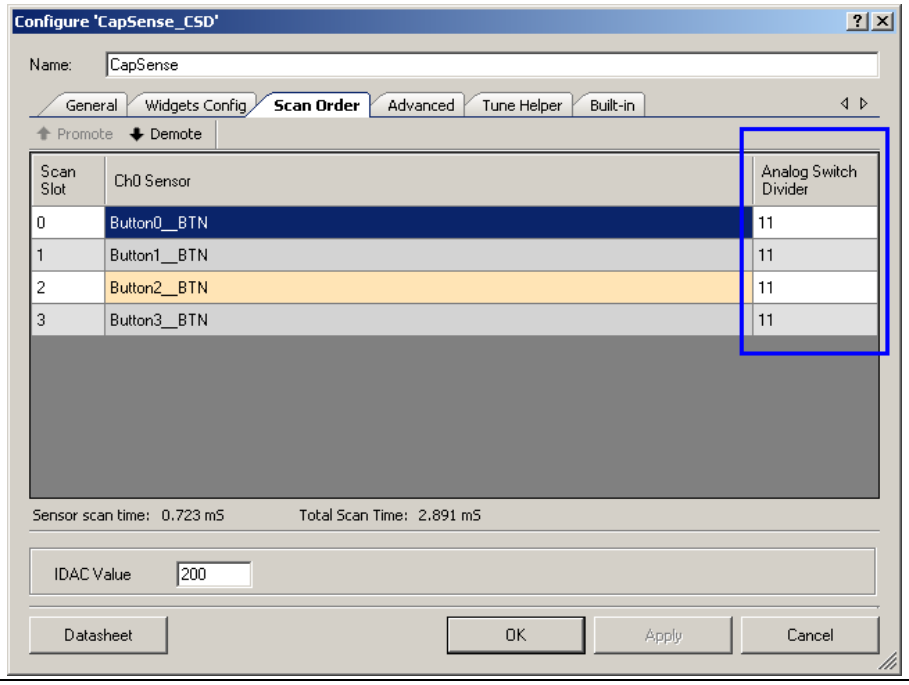

Figure 31 アナログスイッチ分周器パラメーター

アナログスイッチ分周器パラメーターは、CAPSENSE™ センサーのスイッチング周波数を設定します。CAPSENSE™ センサーは、IDAC ソースモードの場合に、C_{MOD} と GND 間を継続的に切り替えます。IDAC シンクと外部抵抗モードの場合に、C_{MOD} と V_{DD} 間を切り替えます。スイッチングクロック周波数は以下のように定義されます。

$$Switching\ Clock = \frac{Scan\ Clock}{Analog\ Switch\ Divider}$$

Equation 5

ここで、

スキャン クロック = CapSense_CSD ブロックへのソースクロック (クロック設定 で説明)

スイッチングクロック = CAPSENSE™ センサーが切り替わる周波数

5.4.2.2 複数のアナログスイッチ分周器

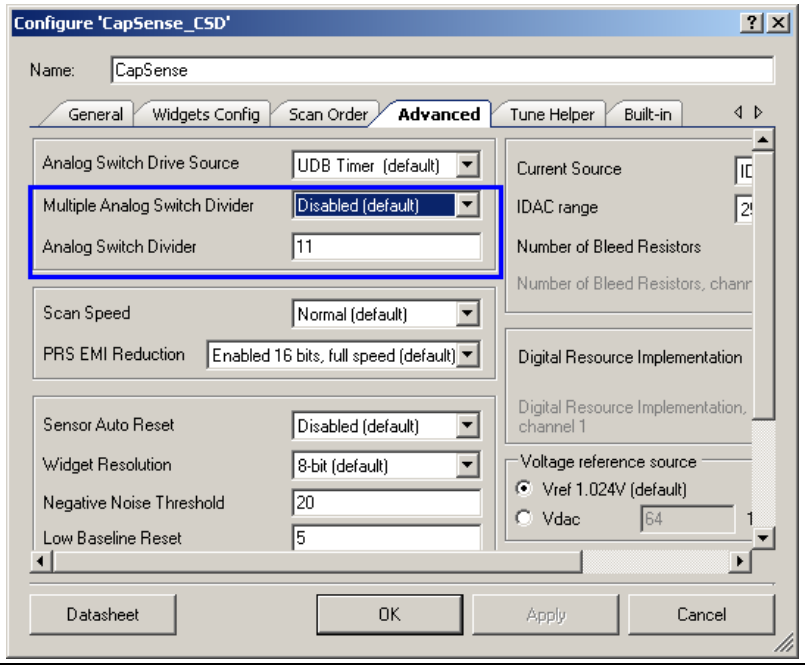

Figure 32 複数のアナログスイッチ分周器

CAPSENSE™_CSD には、各 CAPSENSE™ センサーに対して個別のアナログスイッチ分周器の値を選択するオプション、またはすべてのセンサーに共通の 1 つのアナログスイッチ分周器の値を選択するオプションがあります。すべてのセンサーの C_P が同じになる場合は、共通のアナログスイッチ分周器の値は適切になります。各センサーの C_P 間に大きなばらつきがある場合、複数のアナログスイッチ分周器パラメーターを有効にする必要があります。複数のアナログスイッチ分周器パラメーターを有効にした場合、Figure 32 に示したように、その値は Scan Order タブで設定されます。

5.4.2.3 アナログスイッチドライブソース

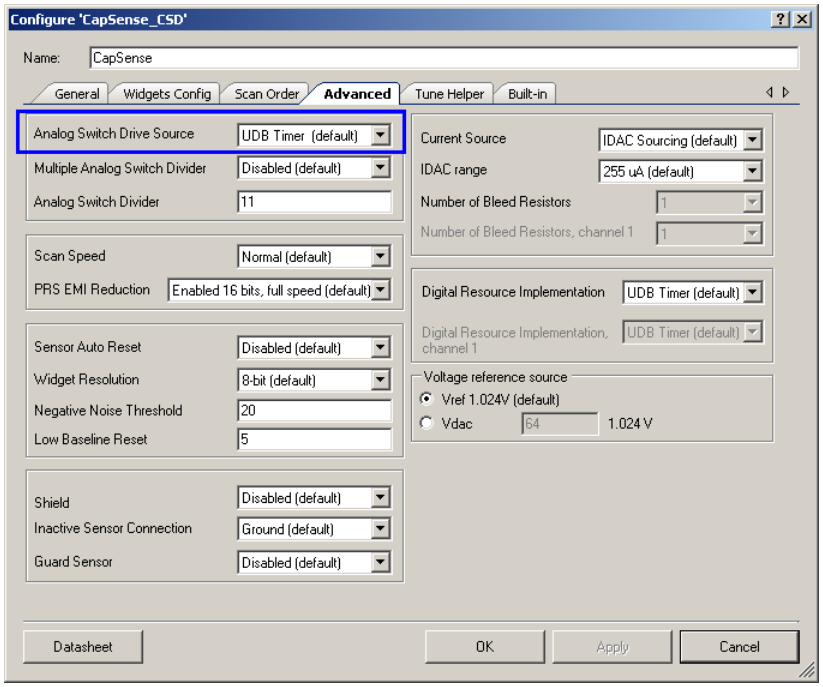

Figure 33 アナログスイッチドライブソースパラメーター

このパラメーターは、アナログスイッチ分周器の実装方法を選択します。アナログスイッチ分周器はタイマベースの分周器です。このタイマの実装に使用されるリソースは次の 3 つのオプションから選択できます。

- **Direct (直接)** - このオプションを選択すると、アナログスイッチ分周器が使用されず、スキャンクロック自身がスイッチングクロックになります。重要なリソースの使用が必要になる場合はこのオプションを選択しますが、設計が正常に実行されていることを確認する必要があります。
- **UDB Timer (UDB タイマ)** - このオプションを選択すると、アナログスイッチ分周器のタイマは UDB を使用して実装します。複数の UDB ブロックがあるため、これはデフォルトオプションです。
- **FF Timer (FF タイマ)** - このオプションを選択すると、アナログスイッチ分周器のタイマは固定機能デジタルブロックを使用して実装します。

5.4.3 擬似乱数シーケンス (RPS)

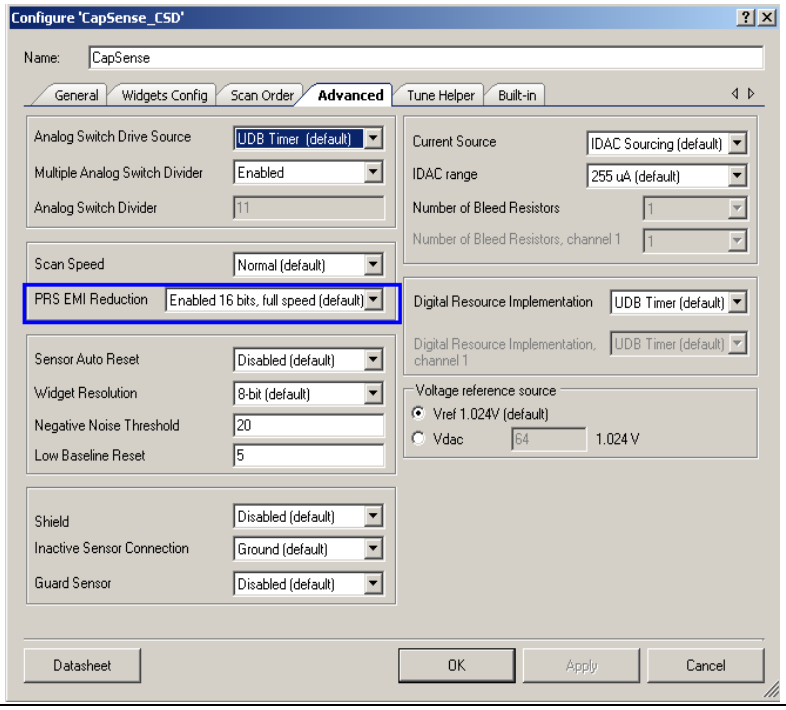

Figure 34 PRS EMI 減少パラメーター

このパラメーターは、スイッチングクロックで疑似乱数シーケンス (PRS) を使用できます。PRS は、スイッチングクロック周波数をその中心の周波数を上回ったり、下回ったりし、そのスペクトルを拡散して EMI を減少させます。次のオプションから選択できます。

- **Disabled (無効)**: このオプションは最高の SNR を提供しますので、アプリケーションに電磁干渉の減少が必要でない場合に推奨します。
- **Enabled 8 bits (有効、8 ビット)**: 8 ビットでは 16 ビットより SNR が改善されますが、繰り返し期間が短くなるため、EMI が増加します。
- **Enabled 16 bits, full speed (default) (有効、16 ビット、フルスピード - デフォルト)**: 16 ビットはより低い SNR をもたらしますが、EMI 減少は優れています。EMI の減少のために、このオプションを推奨します。
- **Enabled 16 bits, 1/4 speed (有効、16 ビット、1/4 スピード)**: Enable 16 bits, full speed より 4 倍速い PRS クロック出力が必要です。

5.4.4 スキャン分解能

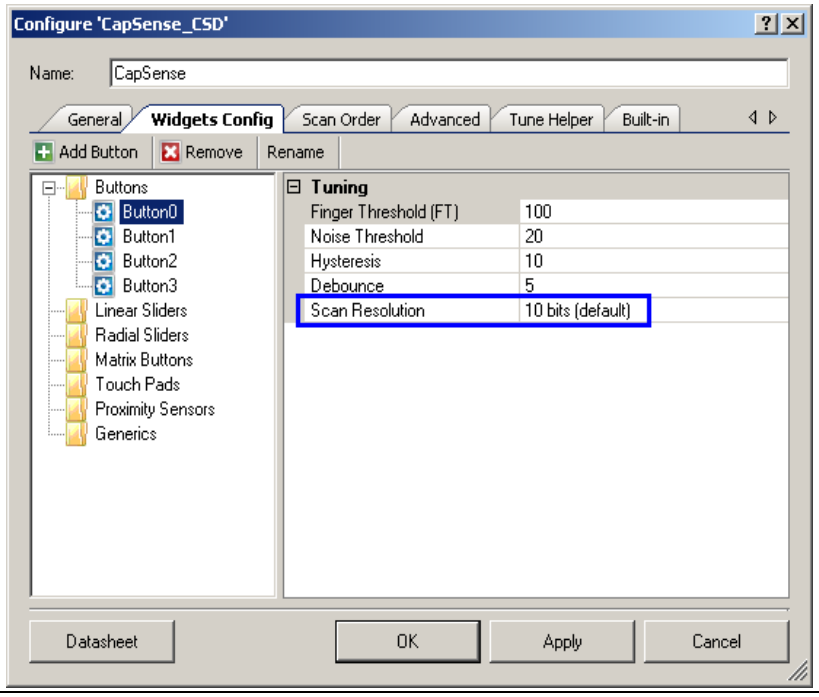

Figure 35 スキャン分解能パラメーター

このパラメーターではスキャン分解能をビット数の単位で設定します。N ビットが 2 のスキャン分解能の場合、最大 raw カウント値は 2^{N-1} です。分解能を高くすると感度が向上しますが、センサーをスキャンするために要する時間も長くなります。使用可能な値は 8～16 ビットです。

5.4.5 IDAC 電流

5.4.5.1 IDAC 値

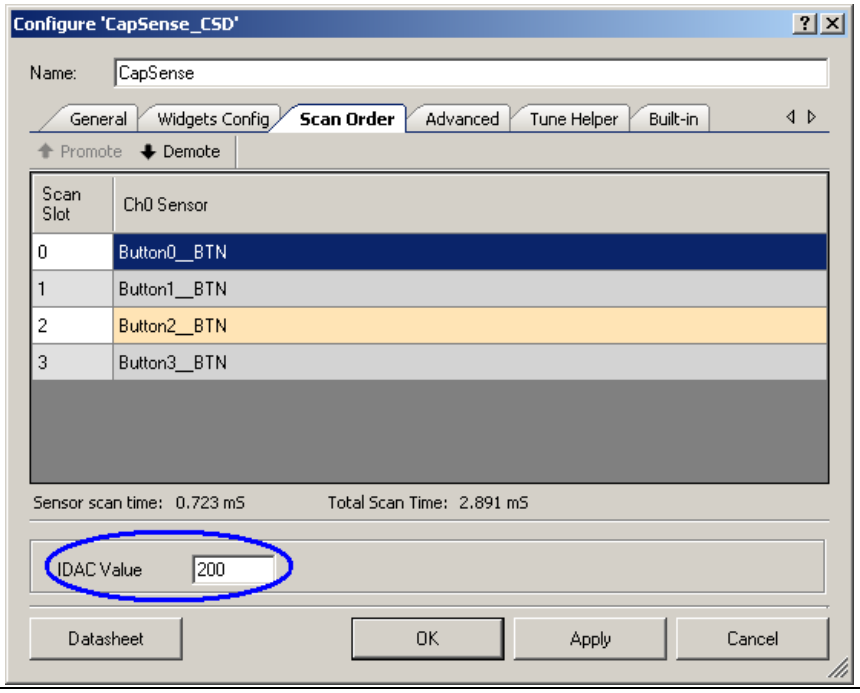

Figure 36 IDAC 値パラメーター

このパラメーターは IDAC 電流を設定します。IDAC 電流が低下すると、raw カウントは感度とともに増加します。N ビットが 2 のスキャン分解能の場合、最大 raw カウント値は 2^{N-1} です。IDAC 電流は、raw カウントが最大値の 50%～ 80%になるように設定する必要があります。

5.4.5.2 IDAC 範囲

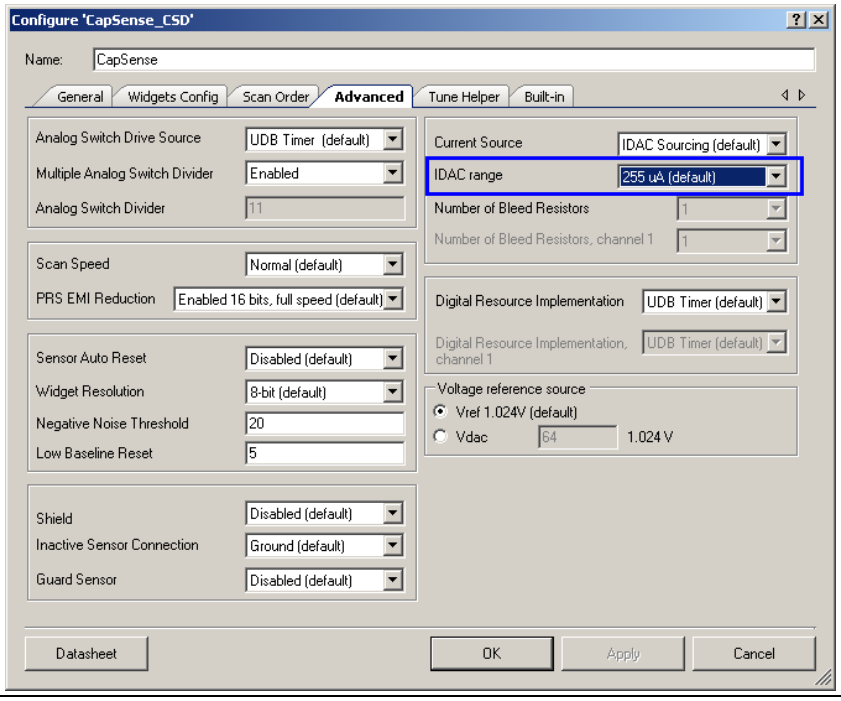

Figure 37 IDAC 範囲パラメーター

このパラメーターは IDAC 電流の範囲を選択します。必要な IDAC 電流は C_P に正比例します。高い C_P の場合は広い範囲を選択し、逆の場合もまた同様となります。使用可能な値は、32 µA、255 µA、および 2048 µA です。

5.4.6 スキャン速度

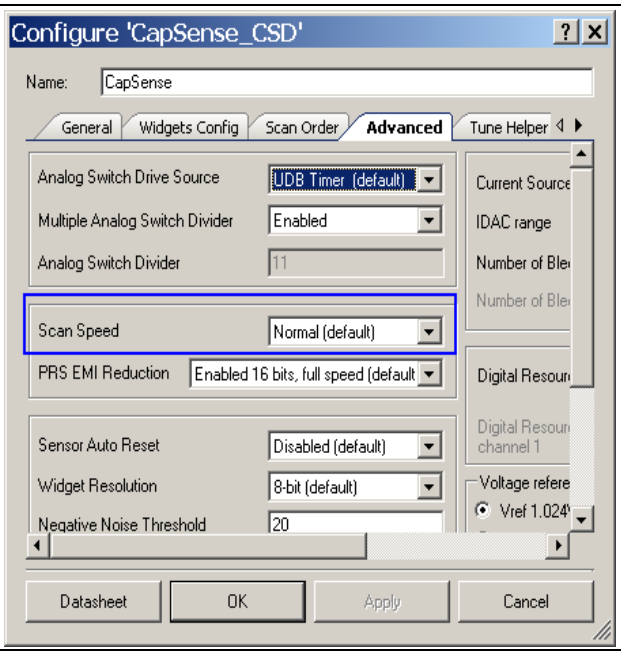

Figure 38 スキャン速度パラメーター

このパラメーターはセンサーのスキャン速度を設定します。速いスキャン速度は短い応答時間をもたらします。スキャン速度が遅いと、SNR が改善され、電源や温度の変化による影響を受けにくくなります。使用可能な値は、Very Fast (非常に速い)、Fast (速い)、Normal (通常)、Slow (遅い) です。

スキャン速度クロックは以下の式でカウントクロックを設定します。

$$Counting\ Clock = \frac{Scan\ Clock}{Scan\ Speed\ Divider}$$

Equation 6

ここで、

スキャン クロック = CapSense_CSD ブロックのソース クロック

スキャン速度分周器 = 1 (非常に速い)、2 (速い)、3 (通常)、4 (遅い)

スキャン時間は、スキャン速度、スキャン分解能、スキャンクロック、および CPU クロックによって異なります。Table 6 に、スキャンクロック = 24 MHz および CPU クロック = 48 MHz で単一のセンサーをスキャンする時間を示します。

Table 6　単一センサーのスキャン時間 (単位: µs) 対スキャン速度と分解能

分解能 (単位:ビット)	スキャン速度			
	非常に速い	速い	通常	遅い
8	58	80	122	208
9	80	122	208	377
10	122	208	377	718
11	208	377	718	1400
12	377	718	1400	2770
13	718	1400	2770	5500
14	1400	2770	5500	10950
15	2770	5500	10950	21880
16	5500	10950	21880	43720

5.4.7 デジタルリソースの実装

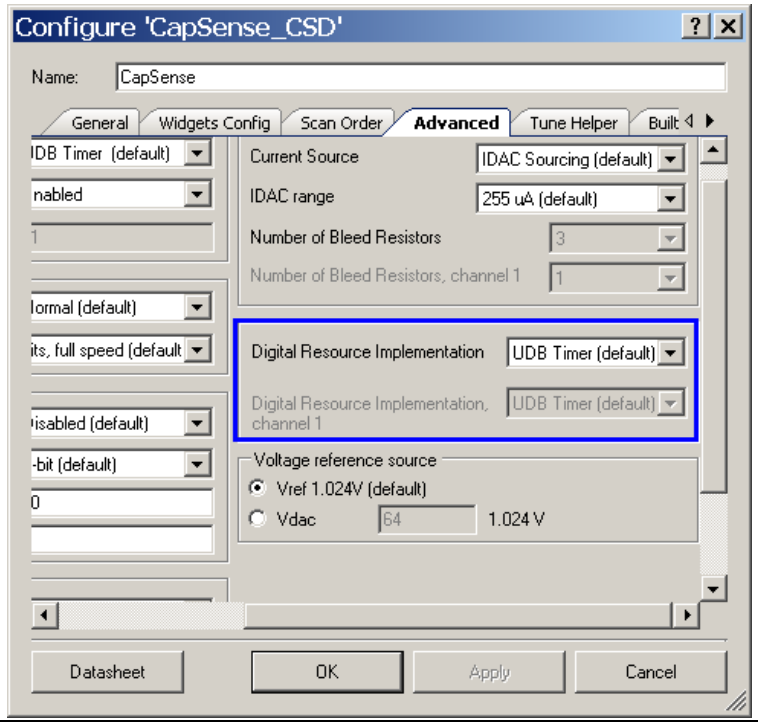

Figure 39 デジタルリソース実装パラメーター

スキャン速度設定は、タイマベースの分周器を使ってカウントクロックを作成します。このパラメーターでは、この分周器の実装用のリソースを選択できます。デジタル リソース実装をチャネルごとに選択できます。オプションは以下のとおりです。

- **UDB Timer (UDB タイマ)**: このオプションを選択すると、タイマは UDB を使用して実装します。複数の UDB ブロックがあるため、これはデフォルトオプションになります。
- **FF Timer (FF タイマ)**: このオプションを選択すると、タイマは固定機能デジタルブロックを使用して実装します。

5.4.8 電流ソース

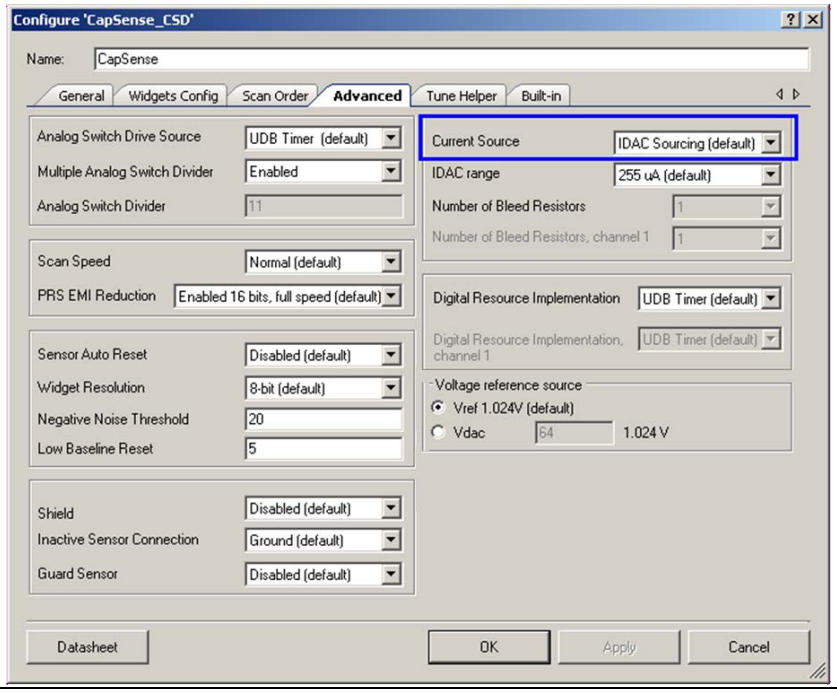

Figure 40 電流ソースパラメーター

このパラメーターは電流ソース方式を選択します。使用可能な値は、IDAC Sourcing (IDAC ソース)、IDAC Sinking (IDAC シンク)、および External Resistor (外部抵抗) です。これらの電流ソース方式の詳細説明は、電流ソース方式 を参照してください。

5.4.8.1 ブリード抵抗数

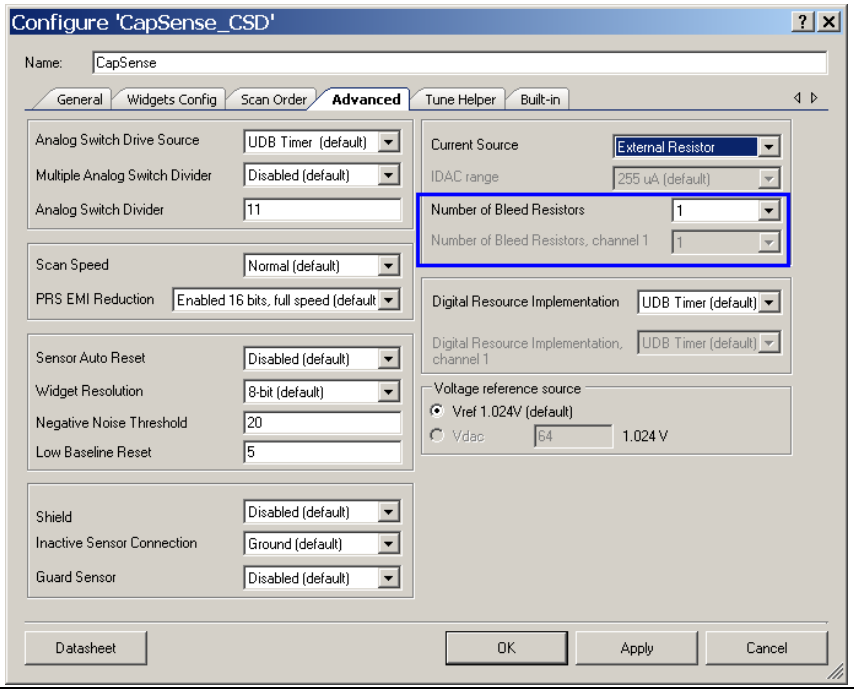

Figure 41 ブリード抵抗数パラメーター

外部抵抗方式は、PSOC™ デバイスの外部ブリード抵抗を必要とします。このパラメーターでは外部ブリード抵抗の数を指定します。最大 3 個のブリード抵抗が接続できます。2 チャネルの静電容量センシングの場合、それぞれのチャネルには個別のブリード抵抗一式が必要です。

5.4.9 電圧基準ソース

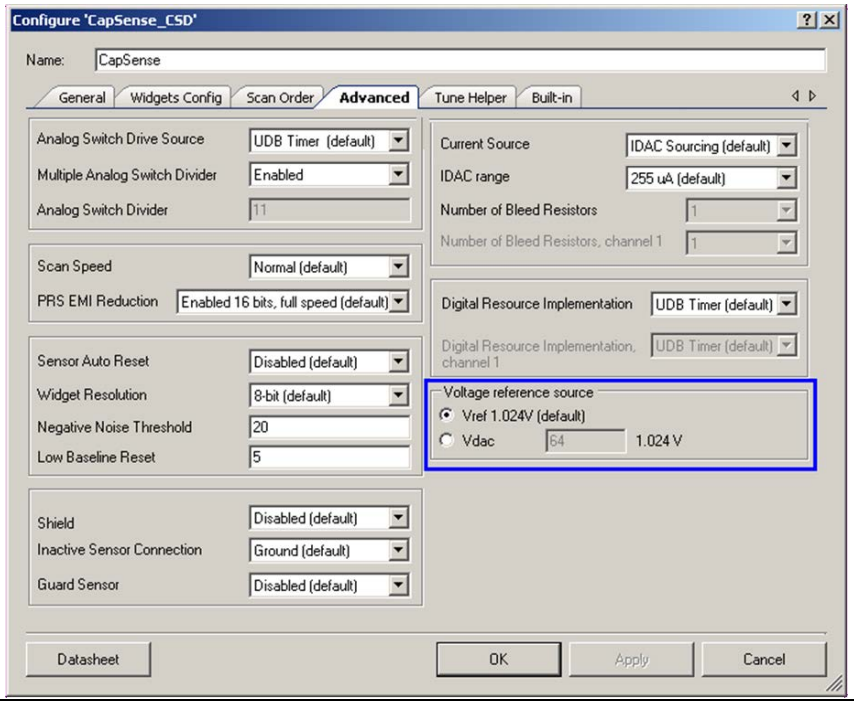

Figure 42 電圧基準源ソースパラメーター

このパラメーターで電圧基準ソースを指定できます。オプションは以下のとおりです。

- **Vref 1.024 V -** これはバンドギャップ電圧です。
- **Vdac -** 0～4V の範囲内のいずれかの電圧を選択できます。IDAC ソース方式を使用する場合に役立ちます。電圧基準を増加させることにより、センサーの電圧振幅を増加させることができます。これは指伝導ノイズに対してより良い耐性を与えます。この設定は 1 つの DAC リソースを使用します。

5.4.10 シールド電極およびガードセンサー

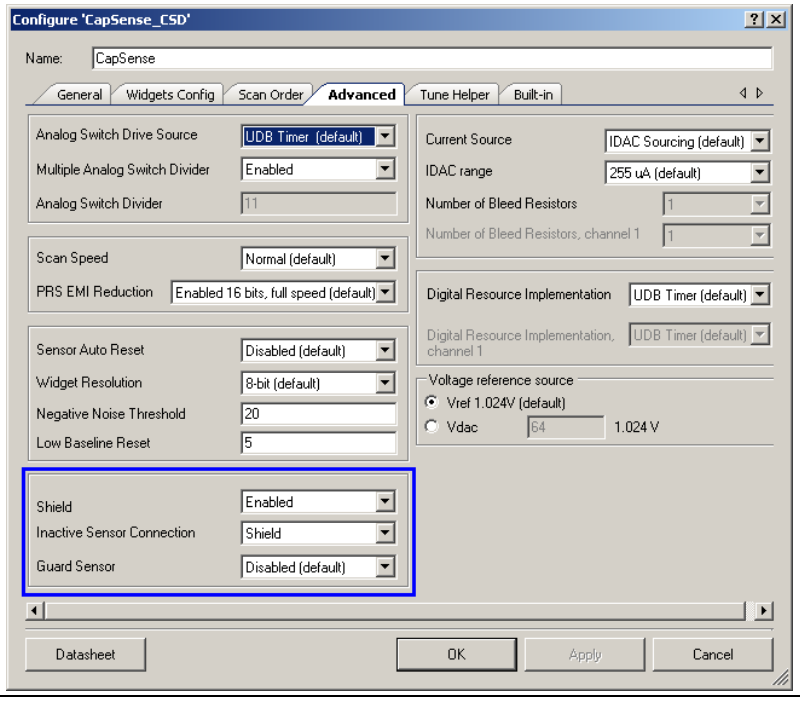

Figure 43 シールド電極およびガードセンサーパラメーター

5.4.10.1 シールド

シールド電極を使用するデザインでは、このパラメーターを有効にする必要があります。耐水性デザイン を参照してください。

5.4.10.2 非アクティブなセンサー接続

このオプションは、センサーがスキャンされていないときのセンサー接続を決定します。使用可能な値は次のとおりです。

- **GND** – この設定はノイズが少なくなるので、推奨されている設定です。
- **Shield (シールド)** – シールド電極を使用し、スライダーなどのセンサーに隣接したウィジェットを含んでいるデザインは、この設定を選択します。非アクティブなセンサーをシールド電極に接続することで、水がセンサー上にある場合の誤トリガーを防ぎます。
- **HI-Z** – 非アクティブなセンサーを高インピーダンスのままにします。この設定を選択すると、センサーに接続されているグランドによる総寄生容量が減少します。

5.4.10.3 ガードセンサー

ガードセンサーを使用するデザインでは、このパラメーターを有効にする必要があります。耐水性デザイン を参照してください。

5.4.10.4 シールド電極用に SIO ピンの使用

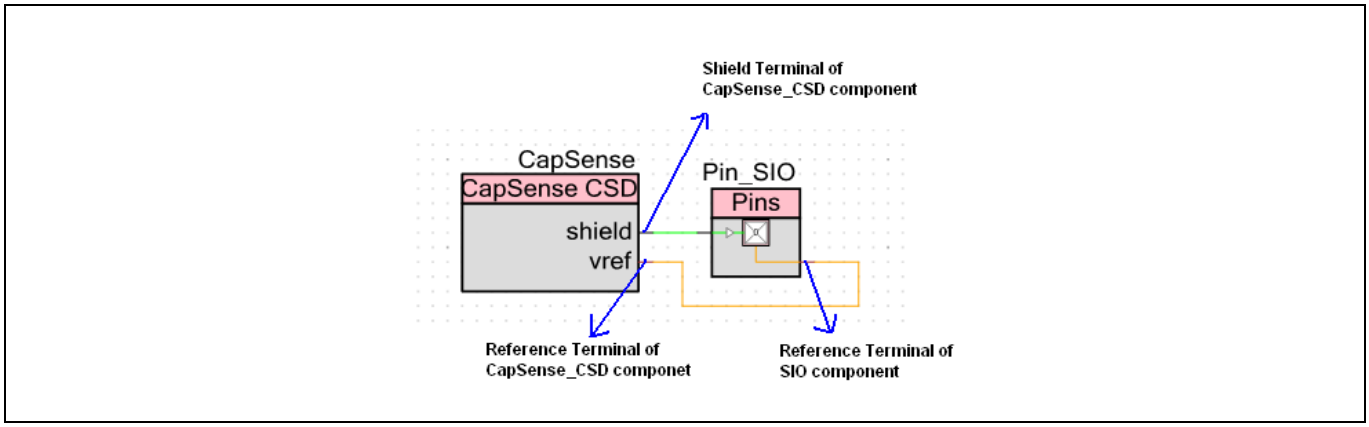

Figure 44 シールド電極で使用される SIO ピン

シールドが有効になると、Figure 44 に示すようにシールド端子がコンポーネントに現れます。

I/O ピンをこの端子に接続する必要があります。電流ソースに IDAC シンクまたは外部抵抗方式を使用する場合、通常の GPIO ピンを使用します。IDAC ソーシング方式を使用の場合は、シールドを SIO ピンに接続する必要があります。SIO ピンを使用して、シールド信号をセンサー信号に整合します。

シールド電極として SIO ピンを使用するには、コンポーネントカタログからデジタル出力ピンをドラッグします。コンフィギュレーションウィンドウを開いて、以下のように「Drive Level (駆動レベル)」オプションを「Vref」に設定します。

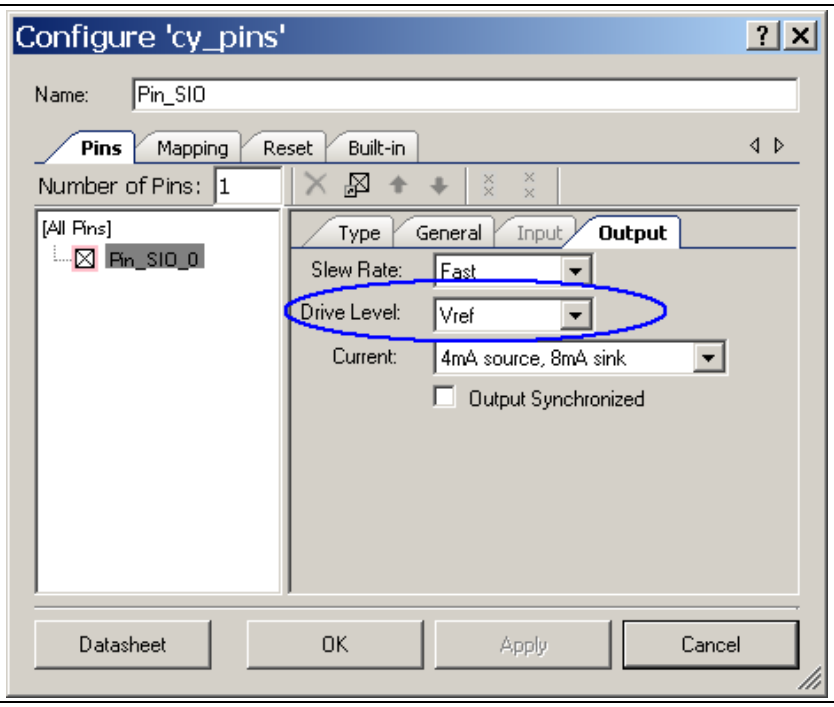

Figure 45 ピンコンポーネントを SIO にするために「Drive Level」を設定

5.4.10.5 複数のシールド電極ピン

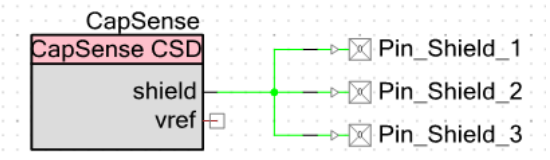

Figure 46 複数のシールド電極ピン

シールド領域が大きい場合、1 本の電極ピンでシールド電極を駆動するのに十分ではないことがあります。シールド電極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。そのために、シールド電極ピンを探り、オシロスコープで監視します。通常のプローブがピンに大きい静電容量を加えることがありますので、10x モードに設定する必要があります。FET プローブを推奨します。シールド電極が完全に充電および放電されない場合は、複数の I/O ピンを使用してシールド電極を駆動します。I/O ピン (GPIO と SIO の両方) には、4 mA の駆動力があります。

5.4.10.6 耐水性および検知

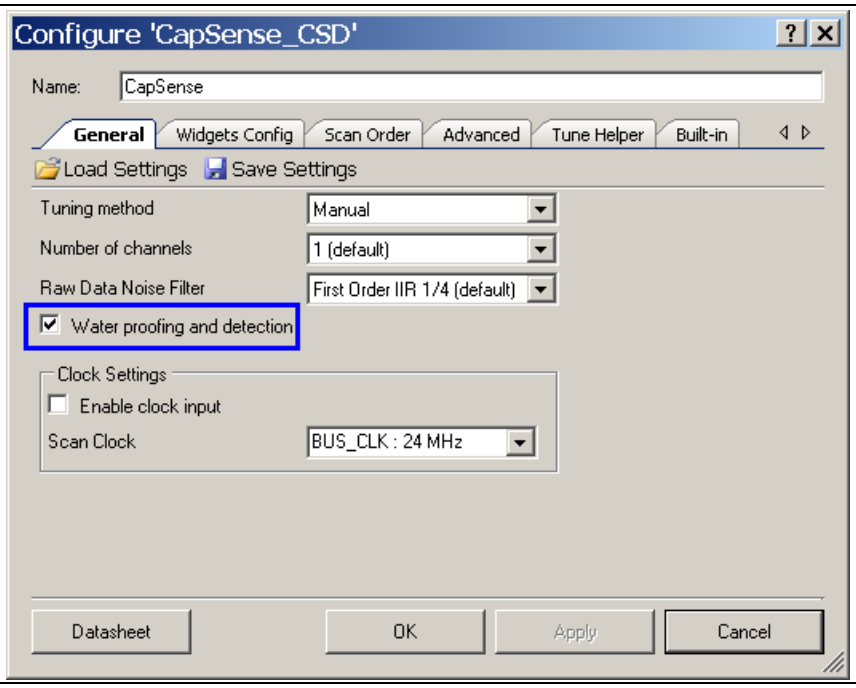

Figure 47 耐水性および検知パラメーター

シールド電極とガードセンサーを有効にするもう 1 つの方法は、「General」タブの「Water Proofing and Detection」パラメーターを選択することです。このボックスをチェックすると、シールド電極とガードセンサーの両方は「Advanced」タブで有効にされます。

5.5 ウィジェットのコンフィギュレーション

「Widgets Config」タブでは、ウィジェットを追加し、設定できます。

5.5.1 ウィジェットの追加

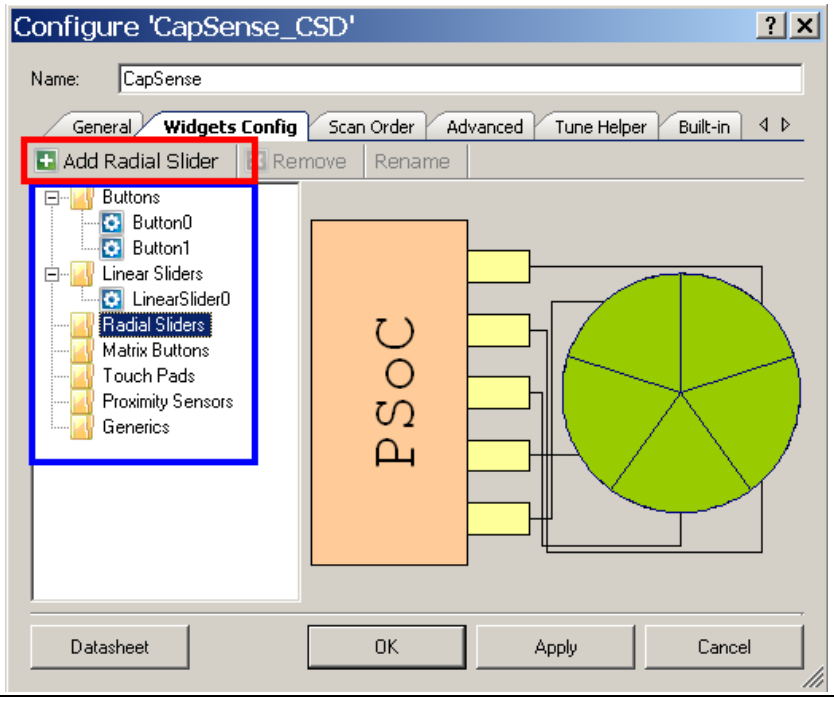

Figure 48 ウィジェットの追加

ウィジェットを追加するには、ウィジェットの種類を選択して、「Add」をクリックします。

5.5.2 センサーエレメント数

各ウィジェットのセンサー数は、このタブで設定されます。

5.5.3 API 分解能

スライダーやタッチパッドなどのいくつかのウィジェットタイプは、センサーのオン/オフ状態の代わりに指位置のデータを使用します。指位置のデータの分解能は、API 分解能と呼ばれます。

5.5.4 近接センサー

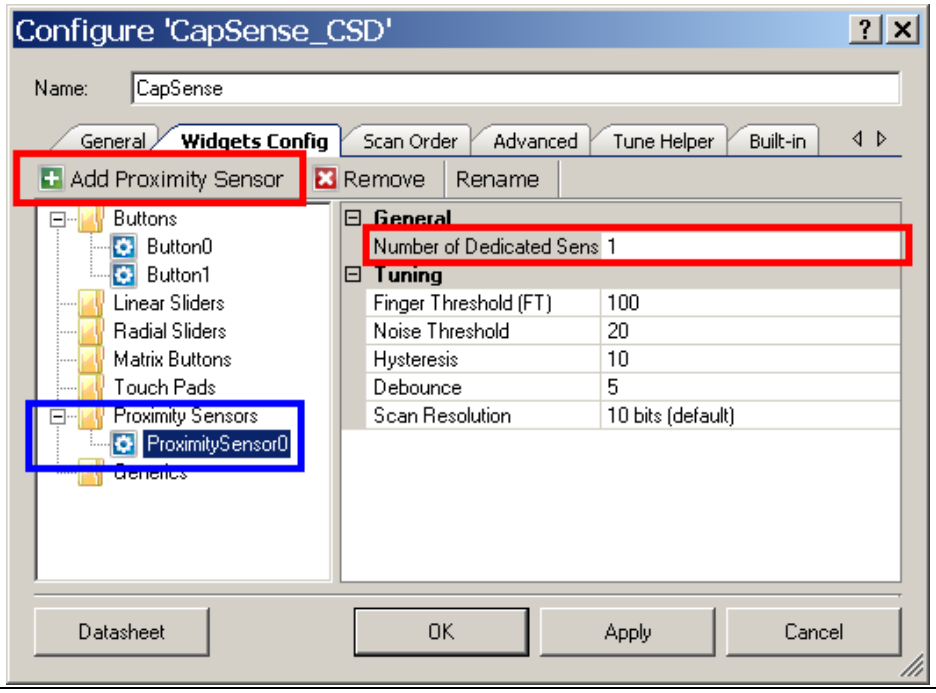

Figure 49 近接センサーパラメーター

近接センサーは、ユーザーがセンサーに触れなくてもセンサー近くの手の存在を検知します。近接センサーは、プリント基板のユーザーインターフェースの周辺に配置している長い配線を含むことがあります。または、センサーを短絡させ、それらを単一のセンサーとしてスキャンすることが可能です。これら 2 つの方法を組み合わせ、ユーザーインターフェースの周辺に配置されている近接センサーの配線を他のセンサーと短絡させ、それらを単一のセンサーとしてスキャンできます。

Note: PSOC™Creator の初期設定では、近接センサーは無効にされ、その他すべてのセンサーは有効にされています。次のコードの一部のように、スキャンする前に有効にする必要があります。

```
17
18      CyGlobalIntEnable;  /* Uncomment this line to enable global interrupts. */
19      CapSense_Start();
20      CapSense_EnableWidget(CapSense_PROXIMITYSENSOR0__PROX); /*Enabling proximity sensor explicitly*/
21      CapSense_InitializeAllBaselines();
22
23      for(;;)
24      {
25          if ( CapSense_IsBusy()==0 )
26          {
27              CapSense_UpdateEnabledBaselines();
28              CapSense_CheckIsAnyWidgetActive();
29              CapSense_ScanEnabledWidgets();
30          }
31      }
32  }
```

5.5.4.1 専用センサーエレメント数

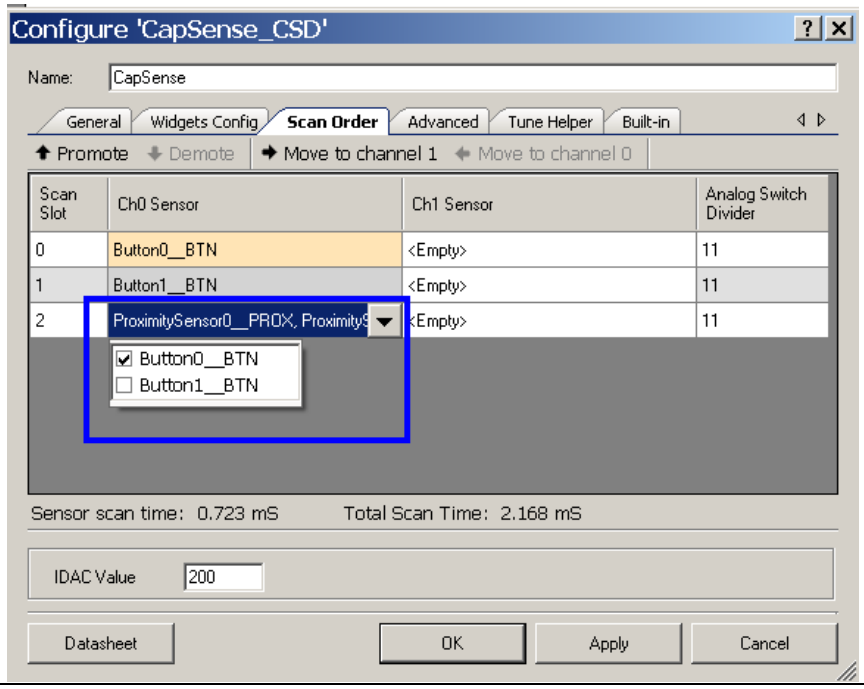

Figure 50 近接センサーと既存センサーの組み合わせ

既存センサーと専用の近接センサーを組み合わせるには、Figure 49 に示すように、Widget Config タブの「Number of Dedicated Sens」を設定します。その後、「Scan Order」タブのドロップダウンメニューで、近接センサーと組み合わせる既存のセンサーを選択します。

既存センサーと近接センサーの組み合わせは、それらの動作に影響を与えませんが、結果として C_P が高く、スキャン時間が長くなります。したがって、長いスキャン間隔を防ぐために、近接センサーは他のセンサーより少ない割合でスキャンする必要があります。

5.6 チューニング方法

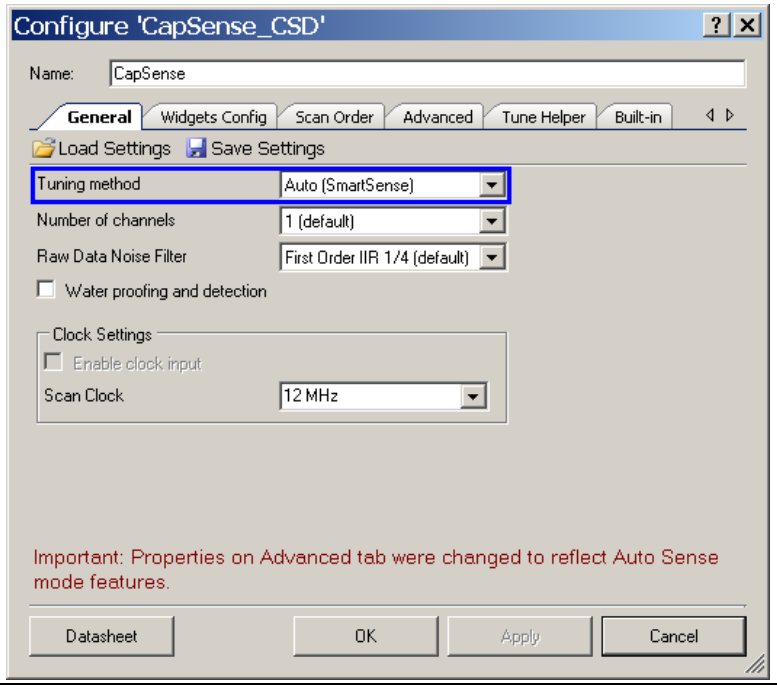

Figure 51 チューニング方法パラメーター

このパラメーターで CAPSENSE™ システムのチューニング方法を選択できます。すべてのチューニング方法では、チューナー GUI を使用して CAPSENSE™ 信号を監視できます。使用可能な値は以下のとおりです。

- **Auto (自動)** – SmartSense 自動チューニングアルゴリズムは、すべてのパラメーターを最適な値に設定します。自動チューニングモードを選ぶと、次のパラメーターを変更できません。

Table 7 自動調整モードの場合の固定パラメーター

パラメーター	設定
電流ソース	IDAC ソース/IDAC シンク 外部抵抗方式は対応していない
IDAC 範囲	255 µA
複数アナログスイッチ分周器	有効
スキャン速度	通常
PRS	16 ビットを有効化、フルスピード

- **Manual (手動)** – すべてのパラメーターを手動でチューニングする必要があります。
- **None (なし)** – パラメーターは固定で、フラッシュに保存されています。手動チューニングか自動チューニングのいずれかの方法を使用して、固定パラメーターをチューニングする必要があります。

5.6.1 感度

感度パラメーターは、SmartSense 自動チューニングアルゴリズムの C_F 値を設定します。感度値は C_F を決めるために 0.1 pF の倍数で設定されます。感度値が 1 の場合は C_F が 0.1 pF に、感度値が 4 の場合は C_F が 0.4 pF になります。

SmartSense 自動チューニングは感度パラメーターを用いて Table 8 に示すパラメーターを計算します。

Table 8 感度を用いて自動チューニングで設定するパラメーター

パラメーター	計算タイミング
アナログスイッチ分周器	CapSense_CSD の起動時に 1 度
IDAC 値	CapSense_CSD の起動時に 1 度
スキャン分解能	CapSense_CSD の起動時に 1 度
指閾値	センサーのスキャン中、継続的に計算
ノイズ閾値	センサーのスキャン中、継続的に計算
ヒステリシス	センサーのスキャン中、継続的に計算

5.7 チャネル数

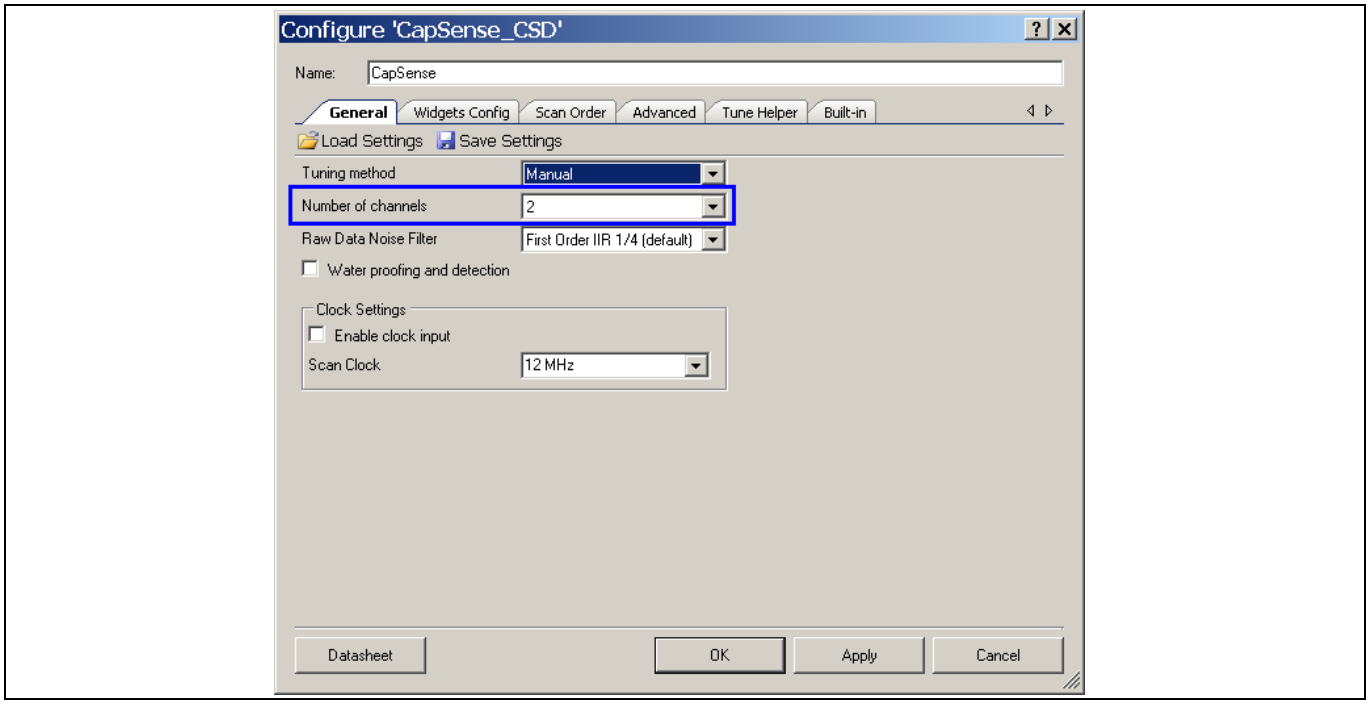

Figure 52 チャネル数

このパラメーターでは 2 チャネル デザインを実装できます。

5.7.1 チャネル 1/チャネル 0 への変更

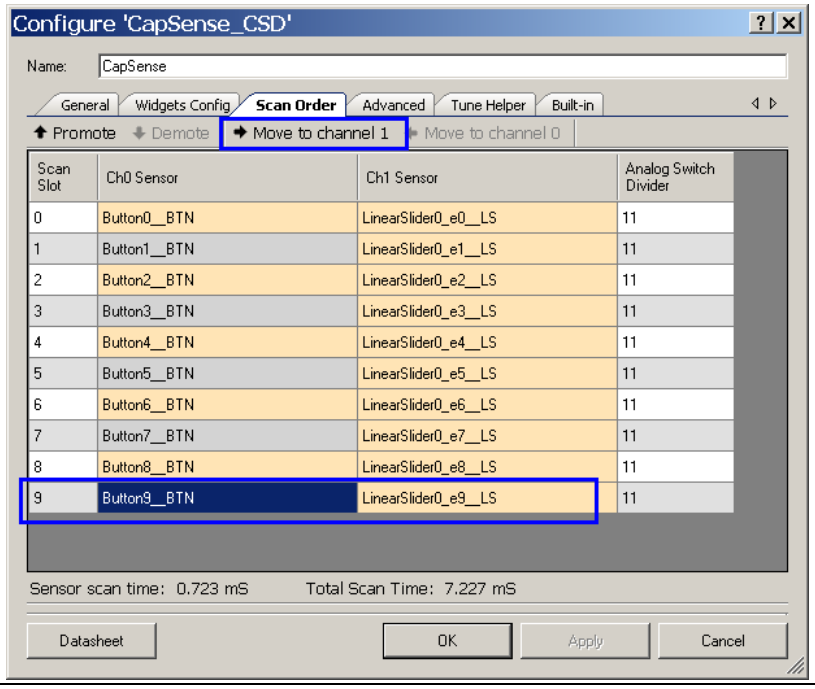

Figure 53 センサーのチャネル割り当ての設定

このパラメーターでは、各センサーにチャネル割り当てを指定できます。「Scan Order」タブで所望するセンサーを選んで、「Move to Channel 1」または「Move to Channel 0」をクリックします。

5.7.2 2 チャネルデザインのピン割り当て

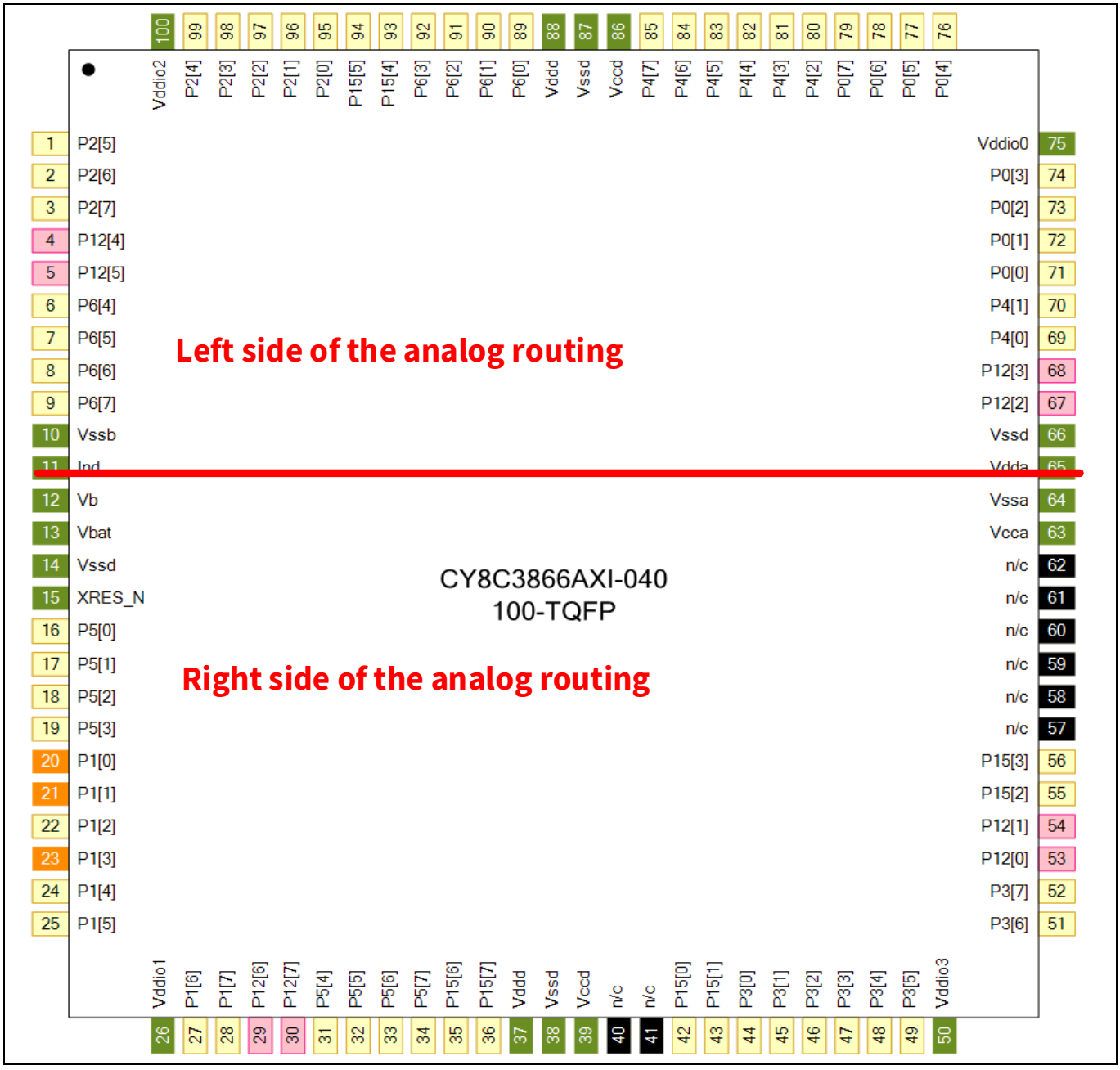

Figure 54 2 チャネルデザインのピン割り当て

PSOC™ の各 GPIO は、内部ルーティングバスとの接続に応じて、アナログルーティングの左側か右側のどちらかの一部です。詳細はデバイスデータシートの「Analog Routing」を参照してください。Figure 54 は PSOC™ のピン配置図であり、GPIO およびそれに対応するアナログルーティングでの側を示します。

2 チャネルデザインでは、1 本のチャネルに割り当てられたすべてのセンサーをルーティングの片側に配置し、もう 1 本のチャネルに割り当てられたすべてのセンサーをルーティングの反対側に配置する必要があります。デザインに 1 本のみのチャネルを使用する場合、C_{MOD} およびブリード抵抗ピンをすべてのセンサーと同じルーティングの片側に配置する必要があります。

5.8 チューナー GUI

CapSense_CSD コンポーネントの便利な機能の 1 つは、CAPSENSE™ 信号を監視する組込み GUI です。チューナー GUI は自動チューニングと手動チューニングの両方に対応しています。CAPSENSE™ 性能のチューニング では、チューニングプロセスを説明します。チューナー GUI は MiniProg3 を介して PC の USB に接続し、I²C を使用してデバイスと通信します。

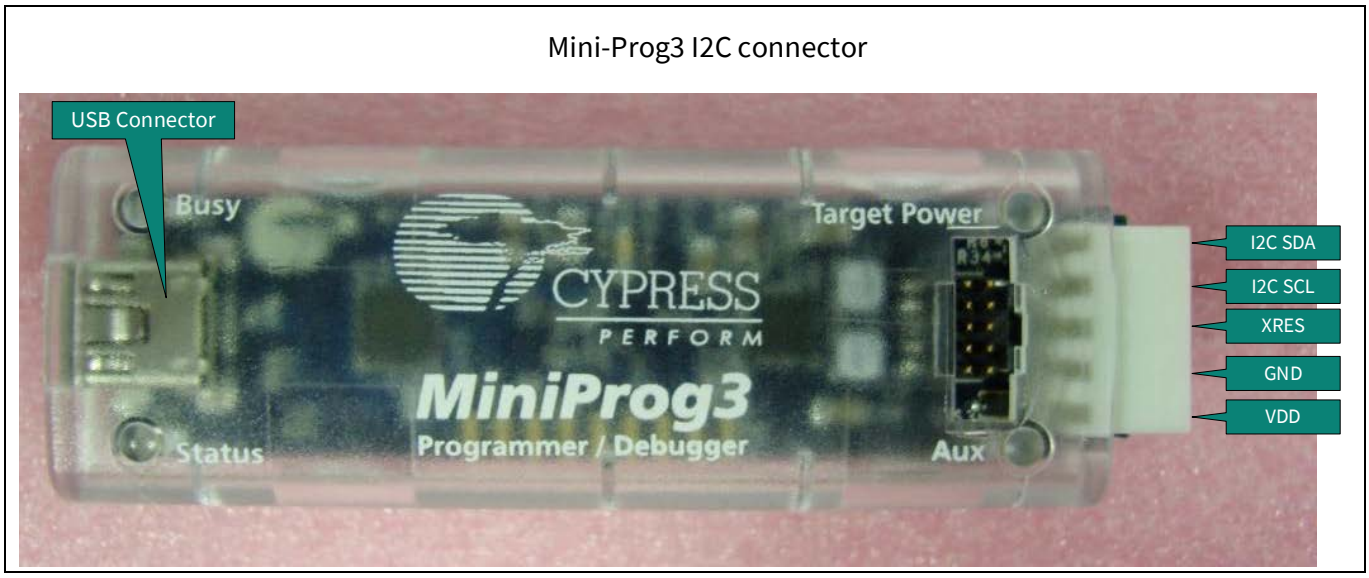

Figure 55 I²C-USB ブリッジとしての MiniProg3

5.8.1 セットアップ

Note: PSOC™Creator プロジェクトをビルドする前に、ステップ1～5 を完了してください。

1. チューニング方法 を自動と手動に設定します。None を選択するとチューナー GUI は動作しません。
2. ウィジェットを追加します。
3. 「Tune Helper」タブで「Enable Tuner Helper」を選択します。

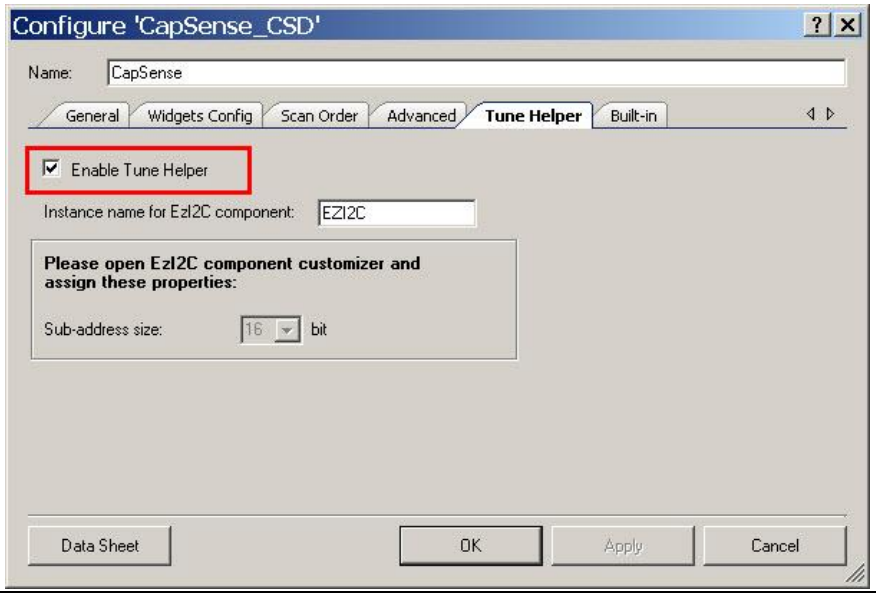

Figure 56 チューナーのイネーブル

4. EZI2C コンポーネントを配置して設定します。

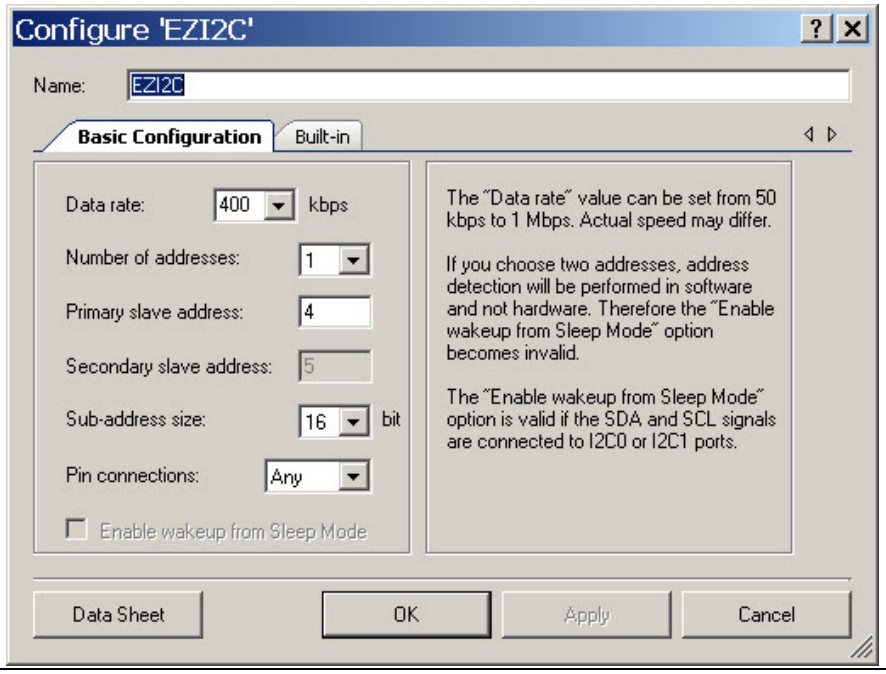

Figure 57　EZI2C コンポーネントの名前を変更

5. 以下のコードセクションをプロジェクトに追加します。

```
     #include <device.h>
void main()
{
CyGlobalIntEnable; /* Global Interrupt enable */

CapSense_TunerStart(); /* CapSense Block and Tuner Communication power up
and Intialization */

While(1)
{
    CapSense_TunerComm(); /*Scan all the sensors and send the result to PC
*/
}
}
```

Note:　デバイスをプログラムして、MiniProg3 を正しく接続させた後に、以下の手順を実行してください。

1. CapSense_CSD コンポーネントを右クリックして、「Launch Tuner」をクリックします。
2. チューナーのウィンドウで、「Configuration」をクリックします。

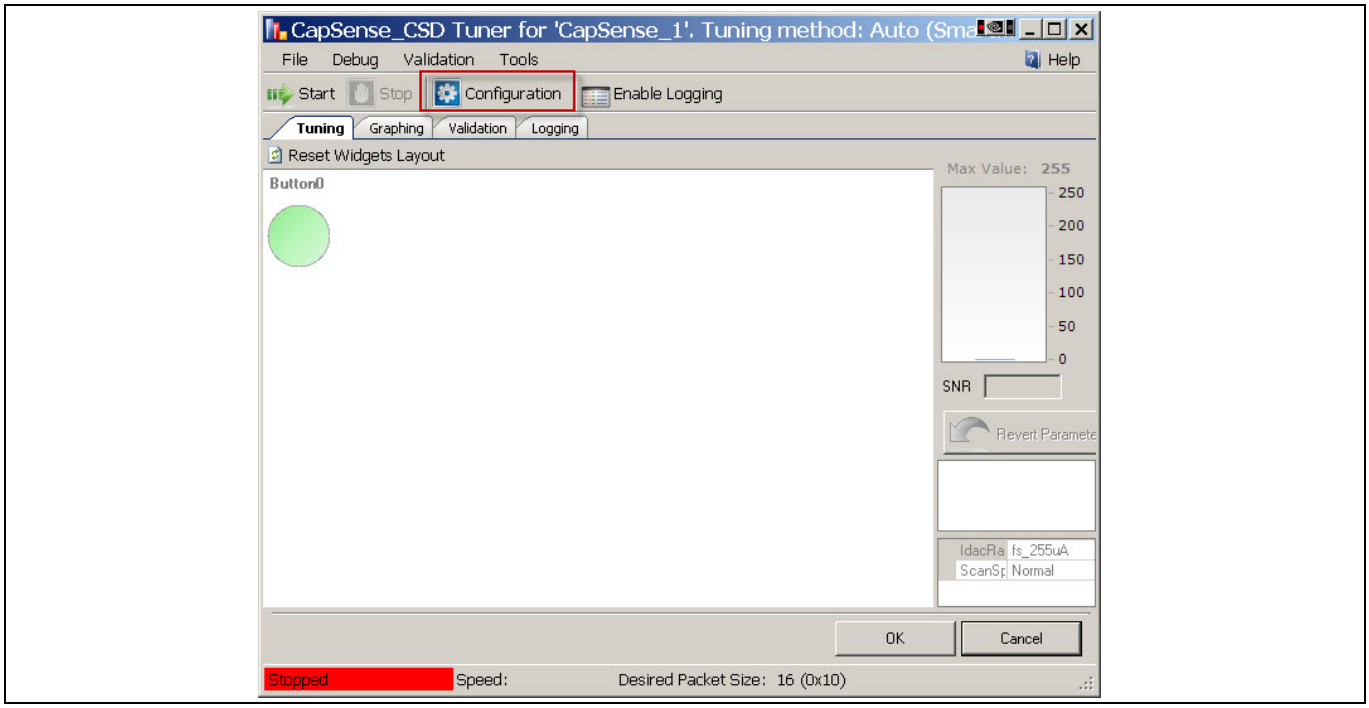

Figure 58　チューナー GUI

3. 「Tuner Communication Setup」ウィンドウで MiniProg3 をクリックして、下図に示すようにパラメーターを設定します。

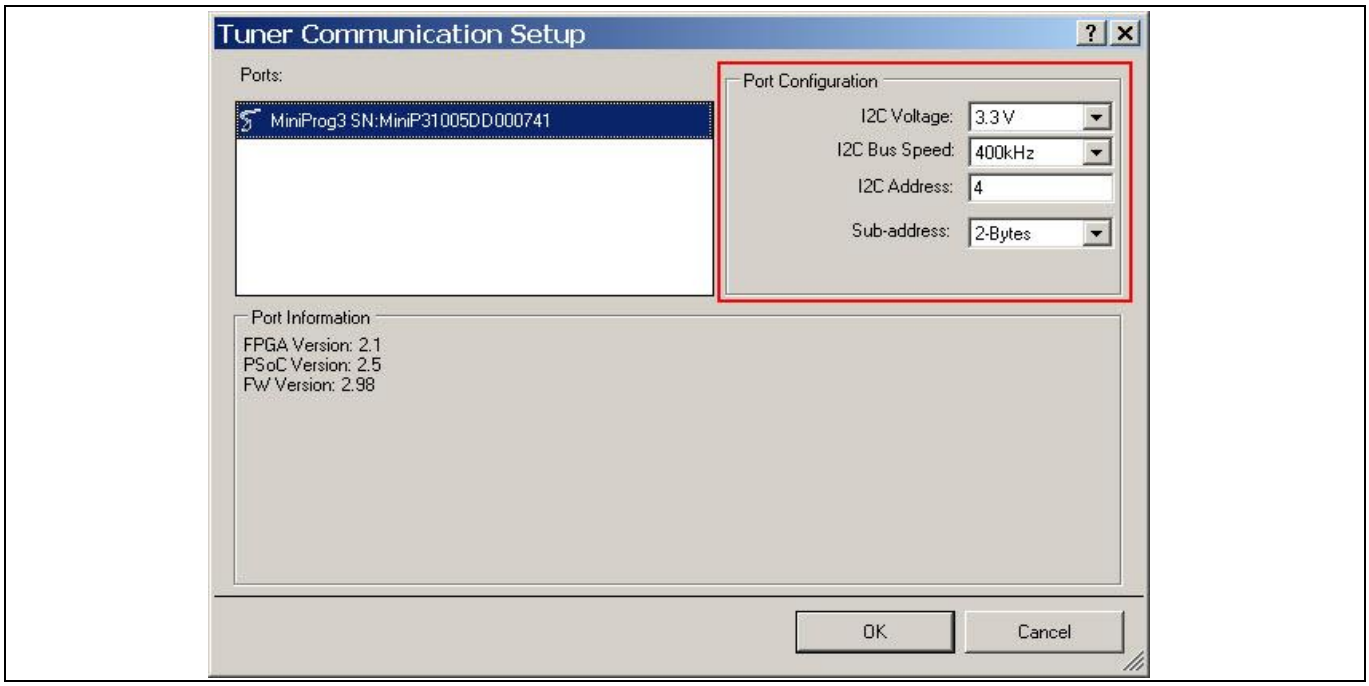

Figure 59　Tuner Communication Setup ウィンドウ

Note:　MiniProg3 の電圧設定は、基板上の電圧設定と一致する必要があります。

4. 「OK」をクリックすると、「Tuner Communication Setup」ウィンドウが閉じます。
5. チューナーウィンドウで「Start」をクリックして I^2C 通信を開始し、CAPSENSE™ パラメーターの監視を始めます。

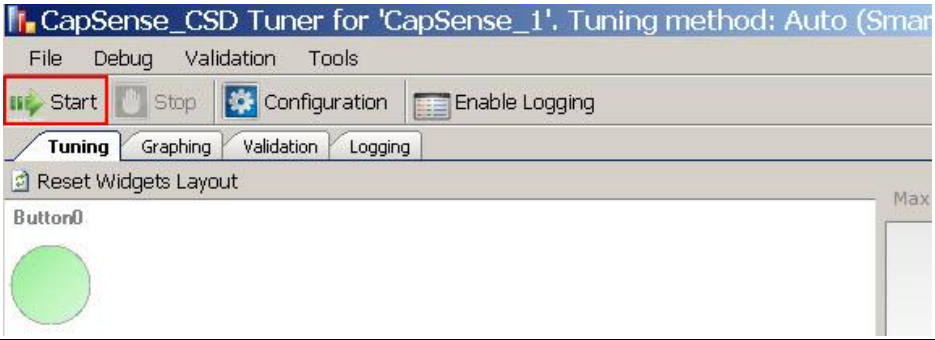

Figure 60　チューナー通信の開始

6. 各センサーのパラメーターを設定します。チューナーウィンドウの右下にある「OK」をクリックした後、チューニングが完了すると、コンポーネントのパラメーターが設定されます。

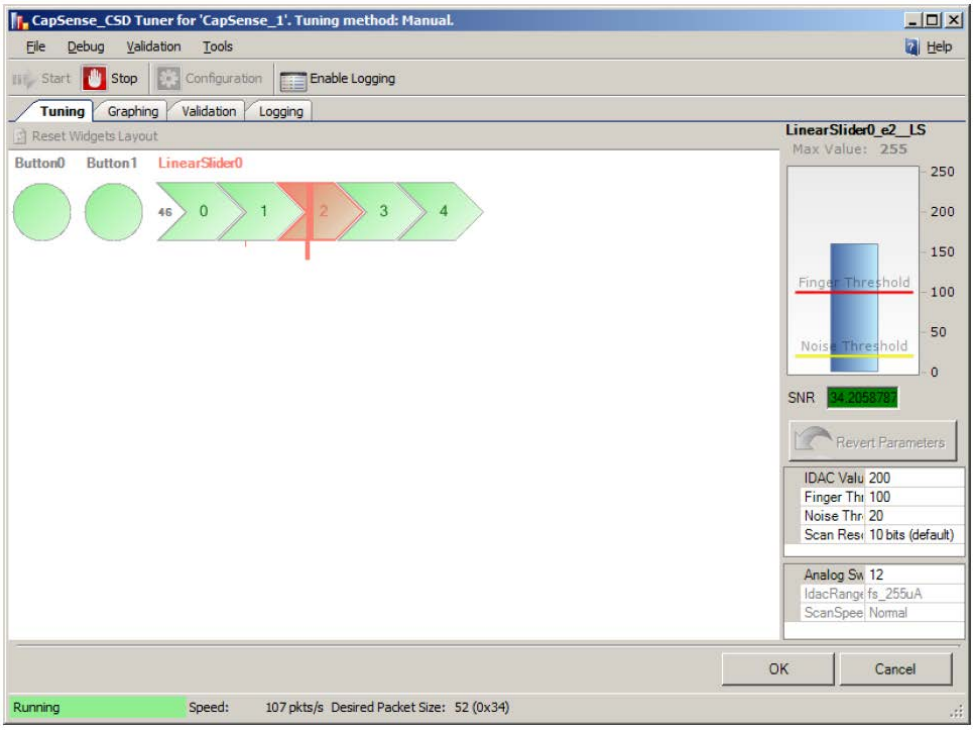

Figure 61　センサーのパラメーターの設定

7. 「Graphing」タブでは、信号を監視できます。データログは「Logging」タブで取得できます。

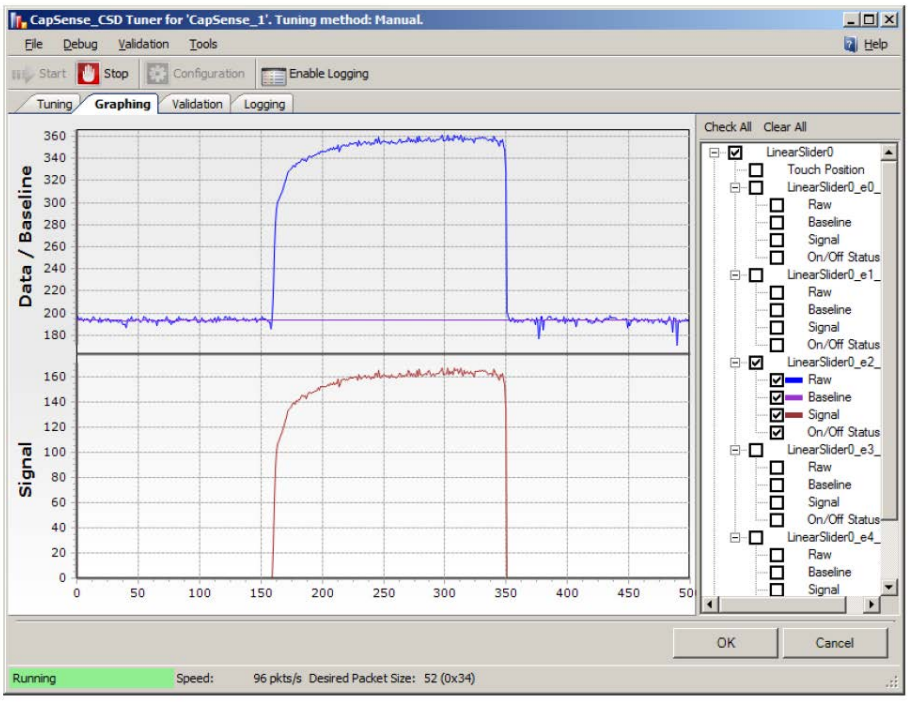

Figure 62 CAPSENSE™結果の監視

Note: 「Graphing」タブに表示されたraw カウント信号はフィルタをかけられていないraw カウントです。raw カウントにフィルタをかけた場合、SNR はフィルタをかけられたraw カウントを使って計算する必要があります。raw データノイズフィルタ を参照してください。

6 CAPSENSE™ 性能のチューニング

チューニングとは、CAPSENSE™ パラメーターの最適な値を決めることです。チューニングは、タッチに対する高い感度を保ち、センサー基板やオーバーレイの素材、環境条件によるプロセスのばらつきを補正するために必要です。

6.1 基本事項

6.1.1 信号、ノイズおよび SNR

適切に調整された CAPSENSE™ システムでは、オンとオフのセンサー状態が確実に識別されます。この性能レベルを得るには、CAPSENSE™ 信号は CAPSENSE™ ノイズよりはるかに大きくする必要があります。信号とノイズを比較する量が、信号対ノイズ比 (SNR) です。CAPSENSE™ の SNR の意味を説明する前に、まずタッチセンシングにおいて信号とノイズとは何かを定義します。

6.1.1.1 CAPSENSE™ 信号

CAPSENSE™ 信号とは Figure 63 に示すように、センサー上に指を置いたときのセンサー応答の変化です。センサーの出力は、センサーの静電容量を追跡する値を表示するデジタルカウンターです。この例では、センサーに指を置かない場合の平均レベルは 5925 カウントです。センサーに指を置いた場合の平均出力は 6060 カウントに増えます。CAPSENSE™ 信号は指接触によるカウントの変化 (信号 = 6060 – 5925 = 135 カウント) を追跡します。

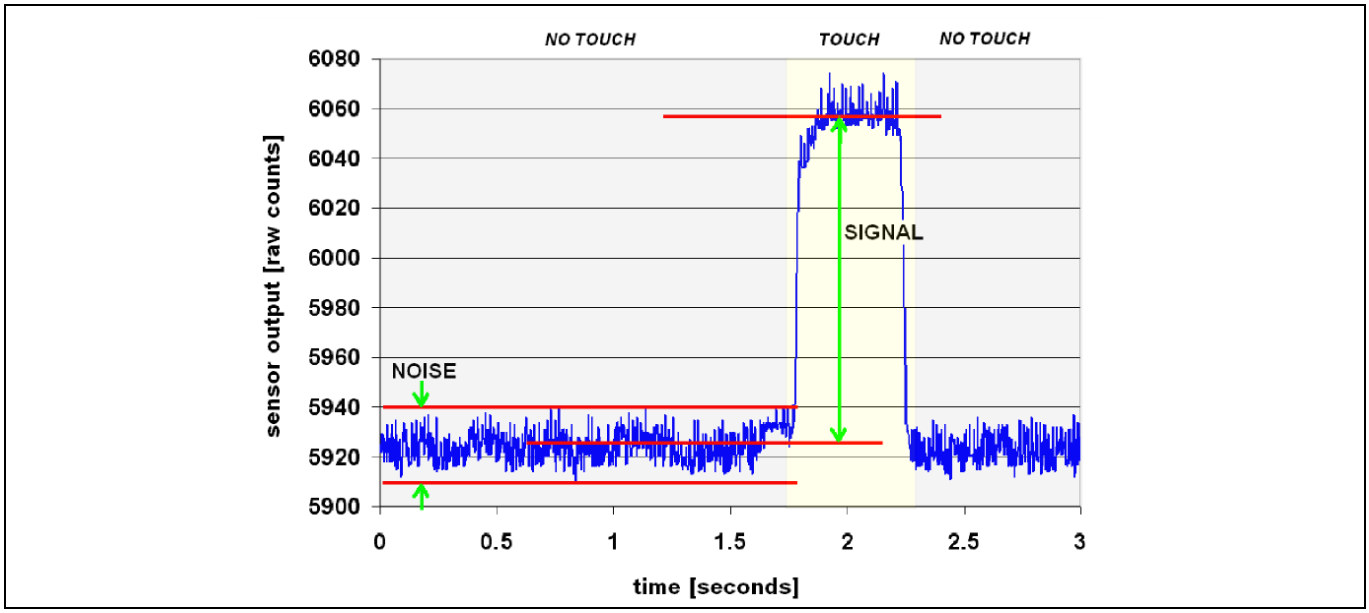

Figure 63 CAPSENSE™ の信号とノイズの例

6.1.1.2 CAPSENSE™ ノイズ

CAPSENSE™ ノイズは、Figure 63 に示すように、センサー上に指を置かないときのセンサー応答のピークツーピーク変化です。この例では、指を置かない場合の出力波形は、5912 ～ 5938 カウントの範囲内に制限されています。ノイズはこの波形の最大値と最小値の差 (ノイズ = 5938 - 5912 = 26 カウント) です。

Note: システムで raw データノイズフィルタ を使用した場合、フィルタされた raw カウントを使ってノイズを計算してください。チューナー GUI は、[Graphing] タブでフィルタをかけていない raw カウントを通知するため、ノイズを計算するためにはデータを記録する必要があります。

6.1.1.3 CAPSENSE™ SNR

CAPSENSE™ SNR とは、信号とノイズの単純比です。この例では、信号が 135 カウントであり、ノイズが 26 カウントである場合、SNR は (135:26) で、簡単にすると (5.2:1) です。CAPSENSE™ の推奨最小 SNR は 5:1 で、すなわち信号値がノイズ値より 5 倍大きいことを意味します。一般的に、フィルタはノイズを減少させるためにファームウェアに実装されます。詳細はソフトウェアのフィルタリング を参照してください。

6.1.2 充電/放電速度

チューニングプロセスにおいて最高感度を達成するには、各サイクル間でセンサーコンデンサを完全に充電および放電しなければいけません。充電/放電パスは、Equation 5 で定義するスイッチングクロックに等しいレートで 2 つの状態間で切り替わります。

充電/放電パスには、電荷移動を減速させる直列抵抗が含まれます。電荷移動の変化速度は、Figure 64 に示すように、センサーコンデンサと直列抵抗から計算する RC 時定数で決まります。

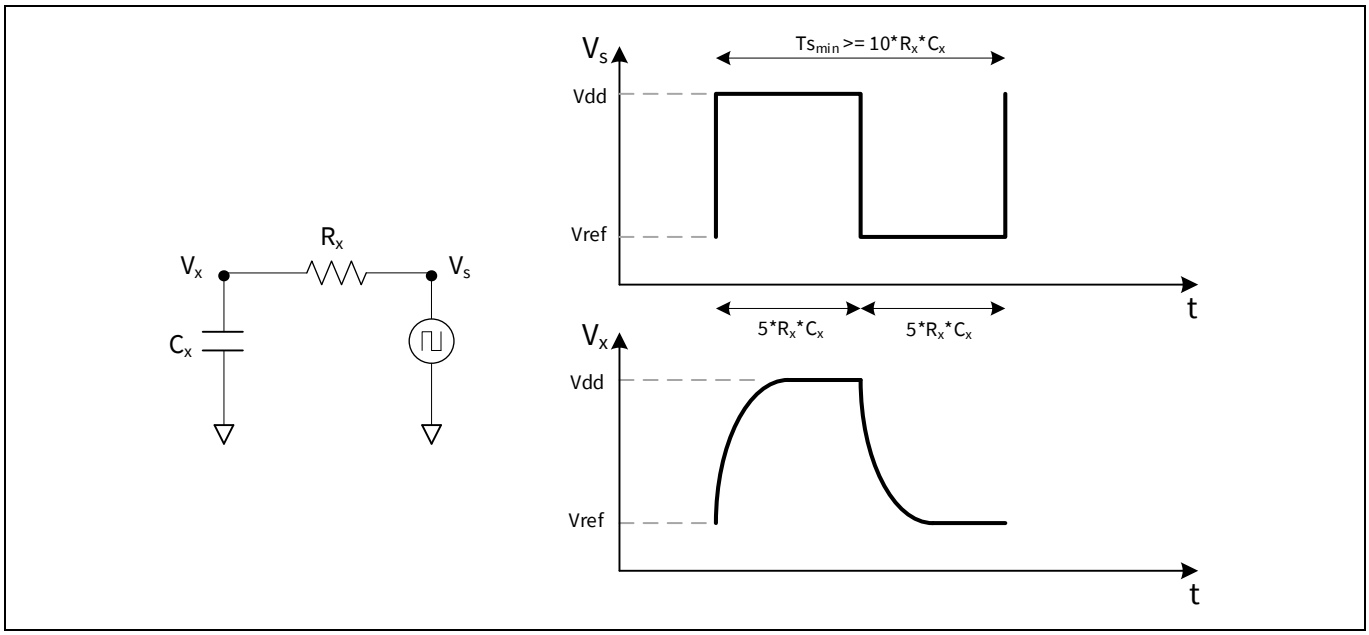

Figure 64 充電/放電波形

充電/放電速度を、RC 時定数に適合するレベルに設定します。経験則として、各遷移には 5RC の期間を合わせて、毎期間 2 回の遷移 (充電 1 回、放電 1 回) とする必要があります。次の式に最小期間と最大周波数を示します。

$$Ts_{min} = 10 \times R_X C_X$$

Equation 7

$$fs_{max} = \frac{1}{10 \times R_X C_X}$$

Equation 8

6.2 手動チューニング

Figure 65 に示すフローチャートは、CapSense_CSD コンポーネントの手動チューニングの手順を提供します。

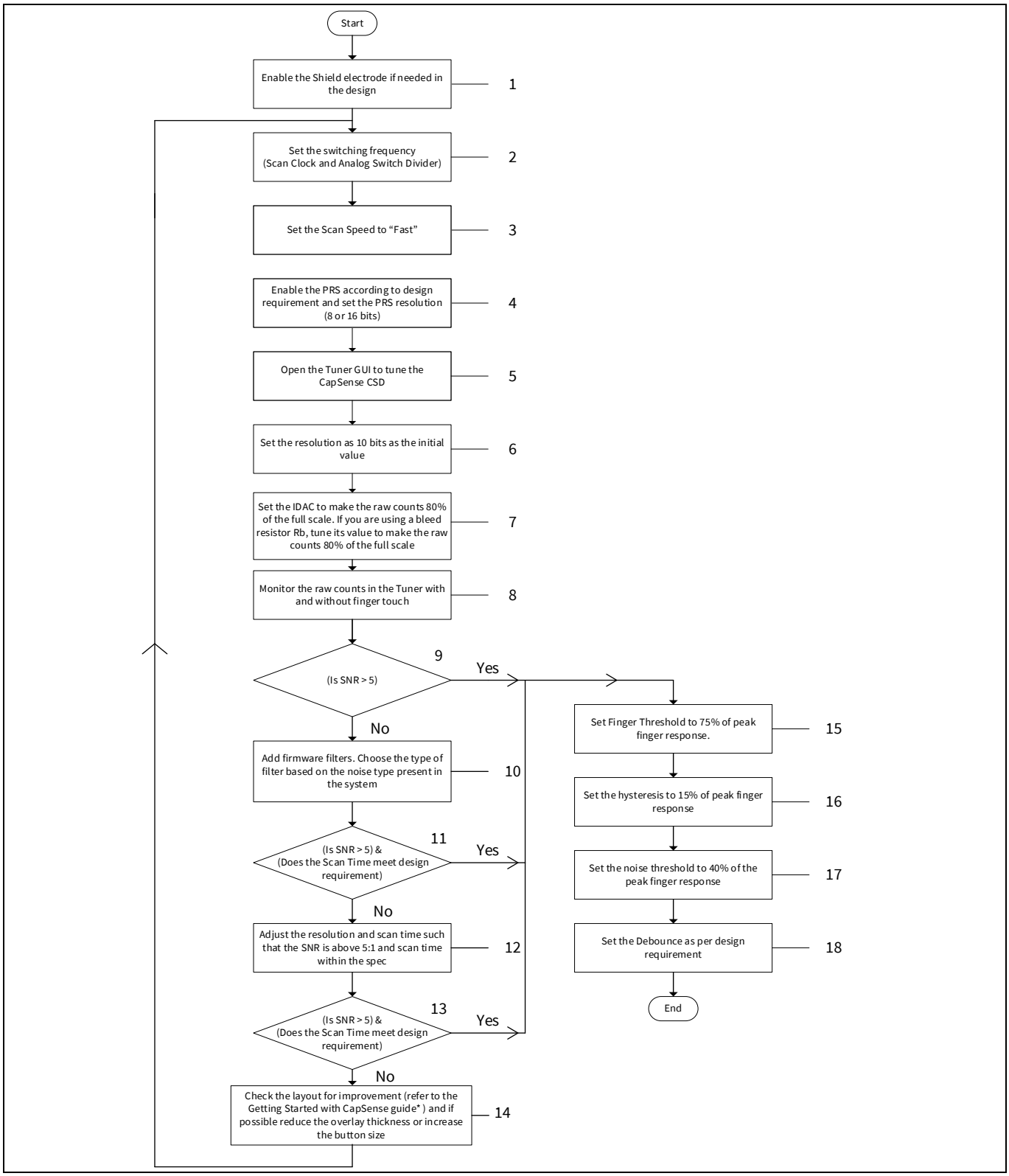

Figure 65 CapSense_CSD コンポーネントのチューニング

1. シールド電極を使用する場合に関する情報は耐水性デザイン を参照してください。
2. センサーが完全に充電および放電されるために、各静電容量センサーのスイッチング周波数はアナログスイッチ分周器 で説明したように設定する必要があります。あるいは、CAPSENSE™_CSD コンフィギュレーションウィンドウでアナログスイッチ分周器の値を増やします。この値を変更すると、プロジェクトをリビルドし、デバイスをリプログラムする必要があります。
3. SNR またはスキャン時間が不適切でない場合、後でスキャン速度を変更できます。
4. この機能をいつ、どのように使用するかについては、擬似乱数シーケンス (RPS) を参照してください。
5. GUI を開くための手順は、チューナー GUI を参照してください。
6. デザインのオーバーレイが 1 mm 未満である場合、より低い分解能 (8 か 9) を初期値として使用できます。
7. raw カウントがフルスケール値の 80% に達するようにチューナー GUI の IDAC 値を調整します。チューナー GUI でフルスケール値の 80% を達成できない場合は、CapSense_CSD コンポーネントのコンフィギュレーションウィンドウで IDAC 範囲を調整します。この値を変更すると、プロジェクトをリビルドし、デバイスをリプログラムする必要があります。
   IDAC シンク/IDAC ソース方式の代わりにブリード抵抗 Rb を使用した場合、raw カウントがフルスケール値の 80% に達するように抵抗値を変更します。
8. ピークツーピークのノイズおよび指のピーク応答を記録します。
9. SNR は単に信号とノイズの比率です。適切な CAPSENSE™ 設計では、SNR は 5:1 以上であることが必要です。この要件を満たした場合、ステップ 15 に進んでください。
10. システムにあるノイズフィルタのタイプを正しく選択するためには、フィルタの選択 を参照してください。First Order IIR 1/4 フィルタは、必要な SRAM を最小限にし、速い応答を提供するため、良い最初の選択となります。
11. SNR が 5:1 以上であることを確認します。そうである場合、デザインの要件を満たすスキャン時間を確認します。スキャン時間を確認するためには、
    - チューナー GUI で OK をクリックして、CapSense_CSD コンポーネントパラメーターを更新します。
    - Scan Order タブを選択して、スキャン時間を確認します

    SNR とスキャン速度の両方の要件が満たされた場合、ステップ 15 に進んでください。
12. SNR を向上するために分解能を増加し、スキャン速度を下げることができます。ただし、両方の変更はスキャン時間を増加させます。
13. ステップ 11 を参照してください。
14. SNR がまだ 5:1 に達していない場合、プリント基板のレイアウトやオーバーレイ設計を改善することを検討してください。オーバーレイ厚さを薄くし、ボタンの直径を大きくすると、感度が高くなります。プリント基板の設計ガイドラインは CAPSENSE™ 入門 の第 3 章を参照してください。
15. SNR を確認するとき、測定した指のピーク応答の 75%を使用します。
16. SNR を確認するとき、測定した指のピーク応答の 15%を使用します。
17. SNR を確認するとき、測定した指のピーク応答の 40%を使用します。
18. デバウンス を参照してください。一般的に、デバウンス値を 2 に設定する必要があります。スキャン速度の高いデザインでは、デバウンス値をより高く (5 などに) 設定してください。

Note: *チューニング方法が「Auto」に設定された場合でも、パラメーターを手動でチューニングできます。チューナー GUI でパラメーターを変更して OK をクリックすると、手動で入力したパラメーターがコンポーネントに適用されます。*

6.3 SmartSense 自動チューニング

チューニング方法に「Auto」を選択すると、SmartSense アルゴリズムは CapSense_CSD コンポーネントパラメーターを選択します。性能を最大限にしながら、CAPSENSE™ の安定した動作を確保するために、SmartSense は各センサーの SNR を (5:1 ～ 11:1) の範囲に維持します。パラメーター設定 で説明した 4 つのみの低レベルパラメーターを設定する必要があります。SmartSense は IDAC シンク/IDAC ソース方式にのみ対応しています。ブリード抵抗方式には対応していません。

6.3.1 ガイドライン

SmartSense 自動チューニングを使用するために、以下のことが必要です。

- すべてのセンサー C_P は 5～45 pF です。
- C_F は少なくとも 0.1 pF です。
- C_{MOD} コンデンサは X7R タイプ, 2.2 nF で、定格電圧が 5 V 以上です。
- SmartSense は手動チューニングと同じ API をサポートしています。ただし、デザイン性能への影響が確かに分からない限り、ファームウェアで CapSense_CSD コンポーネントファームウェアを変更する API を使用しないでください。

6.3.2 パラメーター設定

6.3.2.1 センサー自動リセット

このパラメーターでは、ベースラインが常に更新されるか、または信号差がノイズ閾値以下になったときにのみ更新するかを決めます。Enabled に設定した場合、ベースラインは常に更新されます。この設定では、センサーがオンとなる最大時間を制限します (一般に 5～10 秒) が、システムの故障のためタッチがなくても raw カウントが急に上昇した場合、センサーが恒久的にオンのままになることを防ぎます。センサー自動リセット を参照してください。

6.3.2.2 デバウンス

デバウンスパラメーターでは、デバウンスカウンターをセンサーのアクティブ状態への遷移に追加します。センサーが非アクティブ状態からアクティブ状態になるためには、センサー上で指タッチ信号がデバウンス数の連続したスキャンで検知される必要があります。このパラメーターは、同じ程度ですべてのセンサーへ影響をおよぼします。デバウンス を参照してください。

6.3.2.3 変調器コンデンサピン

このパラメーターでは、C_{MOD} コンデンサが接続するピンを選択します。P0[1] と P0[3] ピンは使用可能です。

Note: SmartSense が正常に動作するためには、2.2 nF の外付けコンデンサが必須です。

6.3.2.4 感度レベル

センサー信号の強さは感度により増減されます。高 (0.1 pF)、中高 (0.2 pF)、中低 (0.3 pF)、低 (0.4 pF) の感度が選択可能です。

より厚いオーバーレイの場合、正常に実装するには、より強い信号が必要です。センサーからより強い信号を生成するために、SmartSense のセンサーをスキャンする時間が増えます。つまり、高感度のセンサーは中高感度のセンサーよりスキャン時間が長くなります。

6 CAPSENSE™ 性能のチューニング

このパラメーターを設定する際、可能な限り低い感度値を使用してください。堅牢な性能を保証するために以下を実施します。

1. チューナー GUI を開きます。
2. 感度レベルを 低 (0.4 pF) に設定して、SNR を計算します。Figure 66 に、指タッチがある場合の一般的な raw カウントグラフを示します。SNR が 5:1 ～ 10:1 の範囲内にあるように感度レベルを調整します。

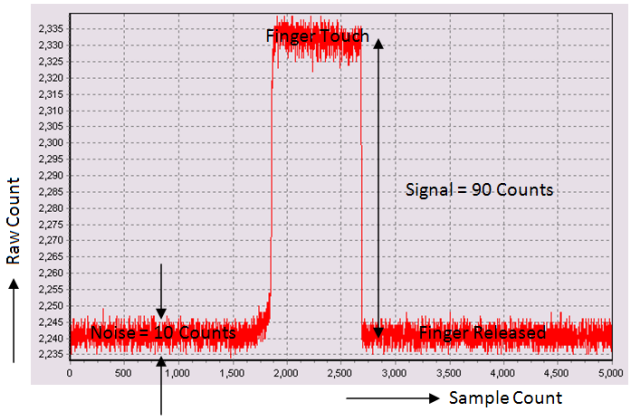

Figure 66　標準センサーへの指タッチがあるときの raw カウントグラフ

7 設計上の注意事項

アプリケーションに静電容量式タッチセンシング技術を組み込む場合、CAPSENSE™ デバイスはより大きなフレームワークを必要とすることに注意してください。プリント基板レイアウトからエンドアプリ環境へのユーザーインターフェースにいたるまで、すべての設計レベルに慎重に設計すると、堅牢で信頼性の高いシステム性能の実現が可能になります。CAPSENSE™ 入門 には、オーバーレイ選択や ESD 保護、電磁環境適合性 (EMC) などの詳しい情報があります。ここでは、PSOC™ 3 および PSOC™ 5LP 固有の設計上の注意事項について説明します。

7.1 ソフトウェアフィルタリング

CapSense_CSD コンポーネントには、CAPSENSE™ のスキャン結果からノイズを除去するために、いくつかの既成ファームウェアフィルタが用意されています。下表では、それらのフィルタおよび使用される場合を説明します。

Table 9 コンポーネント用に用意されている CAPSENSE™ フィルタの表

フィルタ	説明	用途
メジアン	サイズ 3 のバッファからメジアン入力値を計算する非線形フィルタ	モーターやスイッチング電源からのスパイクノイズ
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 (フィードバックなし)	電源からの周期的ノイズ
IIR	RC ローパスフィルタに類似したステップ応答を備えた、有限インパルス応答フィルタ (フィードバック付き)	高周波数ノイズ
ジッタ	前回の入力に基づいて現行の入力を制限する非線形フィルタ	厚いオーバーレイからのノイズ (SNR < 5:1)、特にスライダーの重心データに有用

Note:

1. *コンポーネントのメジアンフィルタはサイズ 2 のバッファを使用します。このバッファには前のサンプルが 2 つ含まれています。バッファに格納されている現時点のサンプルと前の 2 つのサンプルは整えられ、中間値がメジアン値として取得されます。*
2. *コンポーネントの平均フィルタはサイズ 2 のバッファを使用します。このバッファには前のサンプルが 2 つ含まれています。フィルタ出力は、2 つの前のサンプルと現時点のサンプルの平均値です。*

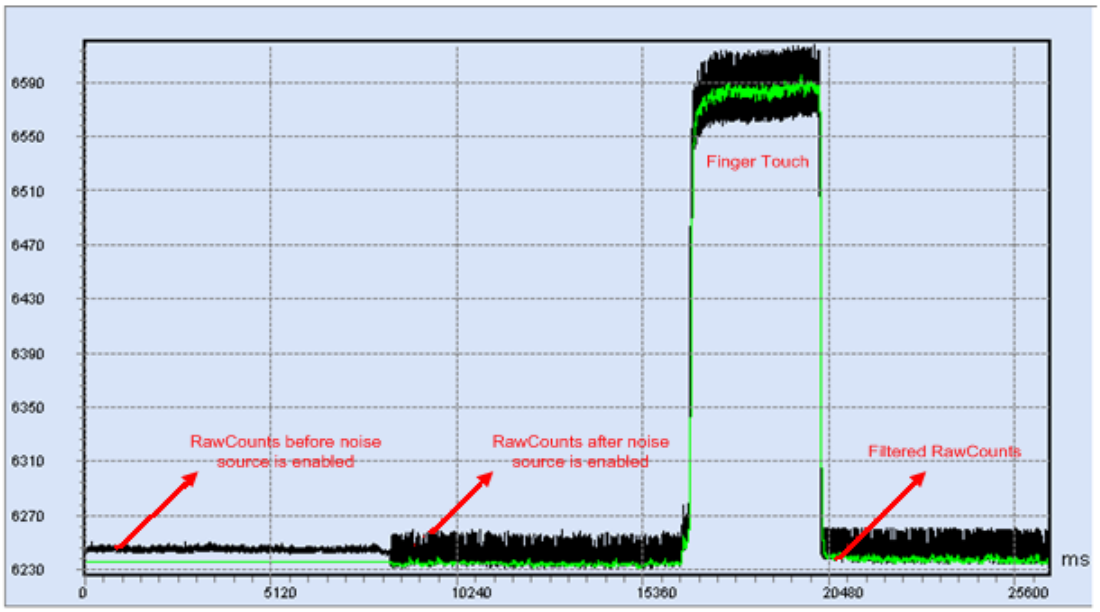

Figure 67 raw カウントのフィルタリング; 黒色 = フィルタをかけられていない raw カウント, 緑色 = フィルタをかけられた raw カウント

Figure 67 に、ノイズとフィルタリングの raw カウントへの影響を示します。センサーの近くの高スイッチングラインによるノイズをシミュレートするために、隣接のセンサーを 500 kHz でスイッチングすることでノイズを生成しました。この例では以下のとおりです。

- ノイズ源のないノイズ = N = 8 カウント (ピークツーピーク)
- ノイズ源のあるノイズ = N1 = 40 カウント (ピークツーピーク)
- フィルタを適用した後のノイズ = N2 = 12 カウント
- 信号 (指のピーク応答) = S = 320 カウント
- フィルタを適用する前の SNR = SNR1 = S/N1 = 320/40 = 8
- フィルタを適用した後の SNR = SNR2 = S/N2 = 320 /12 = 26.7

7.2 消費電力

バッテリ駆動アプリケーションでは、電流レベルを下げると、バッテリ寿命が長くなります。この点を考慮して、システム設計者はシステムの平均消費電流を最小限にする必要があります。

Table 10 PSOC™3 のセンサー 1 個のスキャン用の消費電力

CPU (MHz)	mA (VDD = 3.3 V)	mA (VDD = 5 V)
3	3.7	4.7
6	4.4	5.5
12	5.9	6.9
24	8.4	9.5
48	14.5	16.3

7 設計上の注意事項

CAPSENSE™ 静電容量タッチセンシングシステムの消費電力を削減する方法はいくつかあります。

- **GPIO ドライブ モードを HI-Z に設定**: 初期設定では、すべての GPIO ピンは HI-Z に設定されています。ただし、プロジェクトがいくつかのピンのドライブモードコンフィギュレーションを変更した場合は、それらを使用しないときは HI-Z にする必要があります。
- **低消費電力用に CPU 速度を最適化**: CAPSENSE™ 動作では、専用のハードウェアがセンサースキャンを実施するため、CPU は高速である必要はありません。ただし、CPU はスキャンを開始する前にハードウェアの設定し、ハードウェアがスキャンを完了すると、CPU は結果を処理します。そのため、CPU の速度を遅くすると、スキャン時間が増加します。総スキャン時間を前スキャン、ハードウェアスキャン、後スキャンに分割する方法はノンブロッキング アーキテクチャ を参照してください。
- **低 VDD で動作**

これらの提案に加えて、スリープスキャン方式の適用も効果的です。

7.2.1 スリープスキャン方式

一般的なアプリケーションでは、CAPSENSE™ コントローラーは常にアクティブな状態である必要はありません。デバイスをスリープ状態にして、デバイスの CPU と主要ブロックを停止できます。スリープ状態のデバイスが消費する電流はアクティブ時の電流よりもはるかに小さいです。

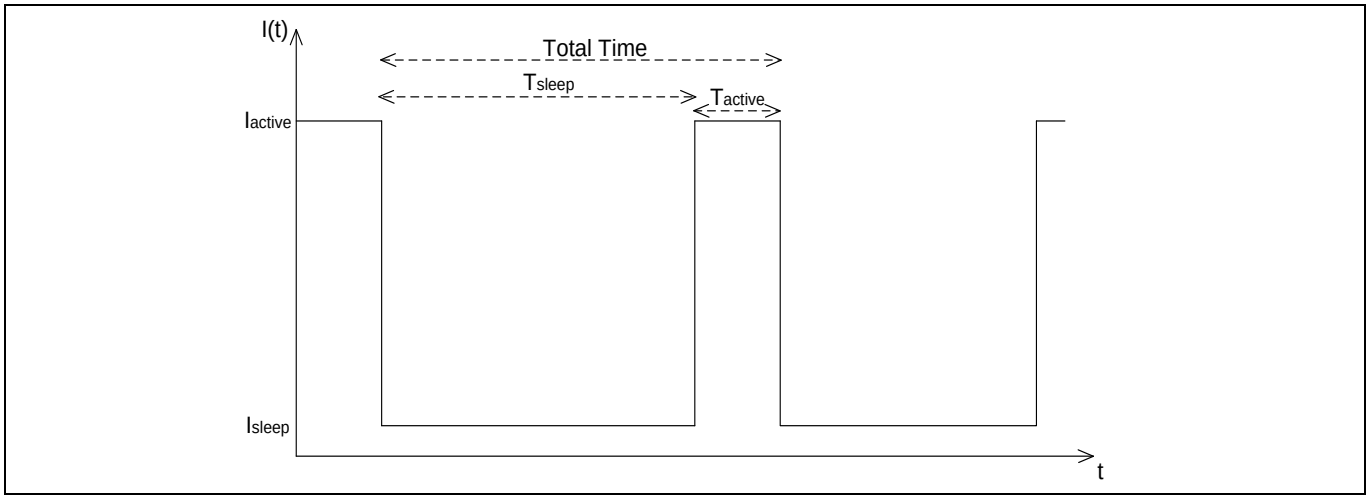

Figure 68 スリープスキャン方式

デバイスの長期的な平均消費電流は、以下の式を使って計算できます。

$$Iaverage = \frac{(Iactive \times Tactive) + (Isleep \times Tsleep)}{Total\ Time}$$

Equation 9

ここで、

$I_{average}$ = デバイスの平均電流

I_{active} = デバイスのアクティブ電流

T_{active} = デバイスのアクティブ時間

I_{sleep} = デバイスのスリープ電流

T_{sleep} = デバイスのスリープ時間

合計時間 = T_{active} + T_{sleep}

デバイスの平均消費電力は、次のように計算します。

$$Paverage = V_{DD} \times Iaverage$$

Equation 10

ここで、

$P_{average}$ = デバイスの平均消費電力

V_{DD} = デバイスの電源電圧

$I_{average}$ = デバイスの平均電流

7.2.2 平均消費電力の測定

7.2.2.1 アクティブ電流

アクティブ電流を測定するためには、デバイスをプログラムしてすべての電源ピン (V_{DDD}、V_{DDA}、V_{DDIO}) の電流を測定します。このために、電源ピンを短絡させて、共通電源ポイントで電流を測定することは最適な方法です。

7.2.2.2 アクティブ時間

スキャンループの前と後に GPIO ピンをトグルすることでアクティブ時間を測定できます。次のプログラムコードは、ピンがスキャンの開始前に設定され、スキャンの完了後にリセットされた一般的な CAPSENSE™ スキャンループを示します。

```
22      for(;;)
23      {
24          if (CapSense_IsBusy() ==0 )
25          {
26              Pin_Write(0);
27              CapSense_UpdateEnabledBaselines();
28              if (CapSense_CheckIsWidgetActive())
29              {
30              /* Do required */
31              }
32
33              /*Check other CapSense sensros active and do the required */
34
35              Pin_Write(1);
36              CapSense_ScanEnabledWidgets();
37          }
38          /* Other application code than CapSense */
39      }
40
41
42
43  }
```

Figure 69 CAPSENSE™ スキャンループプログラムコード

7.2.2.3 スリープ電流

PSOC™ 3 のスリープ電流値は 1 µA、PSOC™ 5LP のスリープ電流は 2 µA です。スリープ電流を測定するために、デバイスを永久的にスリープモードに維持するプロジェクトを作成し、アクティブ電流 で説明した方法を使用します。

7.2.2.4 スリープ時間

API「CyPmSleep()」ではデバイスをスリープモードに移行させます。Figure 70 に示すように、スリープタイマは特定の間隔後に割込みを生成してデバイスをウェイクアップします。

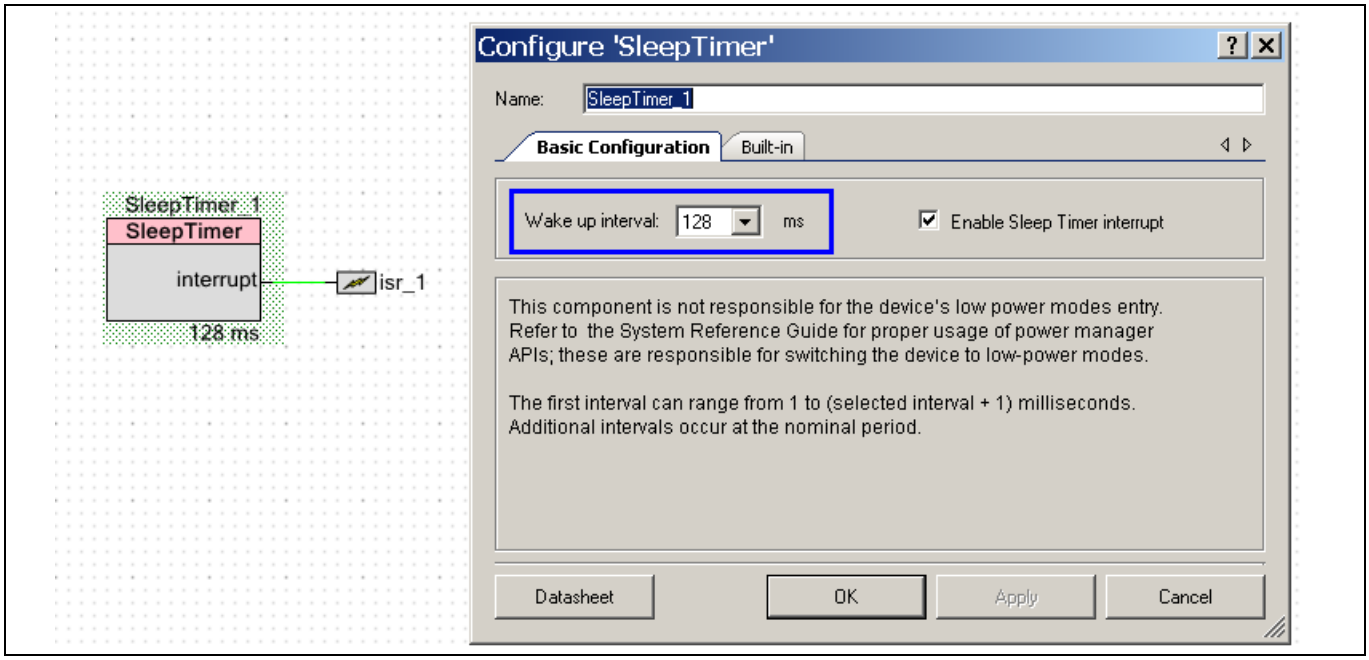

Figure 70 スリープタイマ

7.2.2.5 平均消費電力の計算

アクティブ電流 (Table 9 より、CPU = 24 MHz、動作電圧 = 3 V)

I_{active} = 8.4 mA

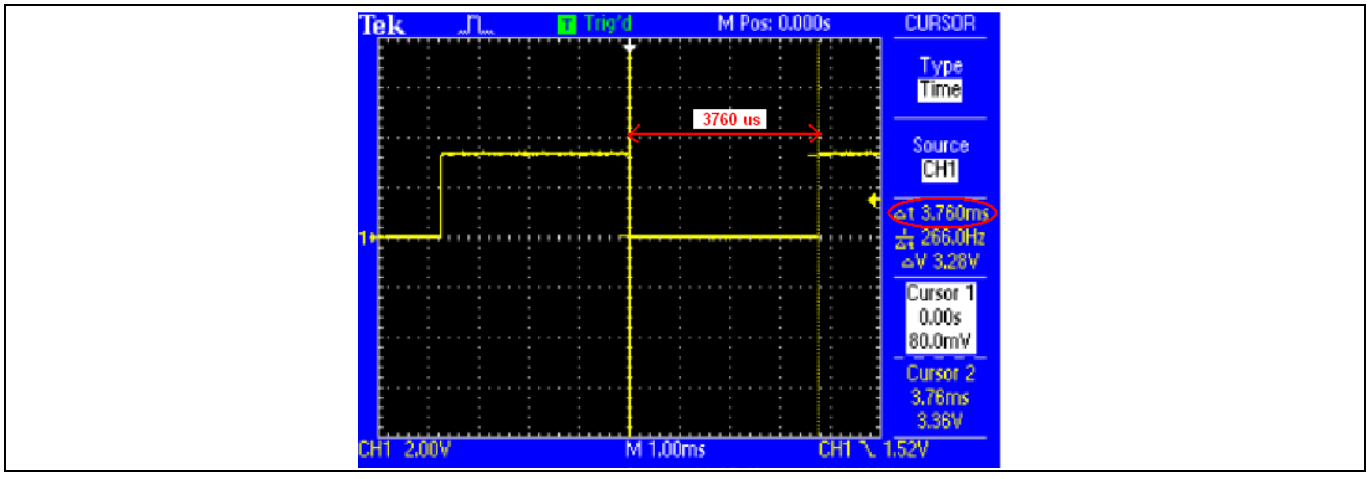

Figure 71 アクティブ時間の測定

アクティブ時間 (PSOC™ 3、CPU = 24 MHz、分解能 = 12 ビット、スキャン速度 = 通常)

T_{active}= 3760 µs

スリープ電流 (データシートより)

I_{sleep} = 1 µA

スリープ時間 (スリープタイマでスリープ期間が 128 ms に設定されたことを想定)

スリープ時間 = 合計時間 - アクティブ時間

合計時間 = スキャン間隔 = 128 ms

T_{sleep} = 128,000 µs – 3760 µs = 124240 µs

平均電流 (Equation 9 より)

$I_{average}$ = 247.7 µA

平均消費電力

$P_{average} = V_{DD} x\ I_{average}$

$P_{average}$ = 3.3 V x 247.7 µA = 817.4 µW

スリープスキャンモードを使用する平均消費電力は 817.4 µW になります。一方、スリープモードを使用しないシステムの平均消費電力は 27.7 mW になります。

7.3 応答時間

7.3.1 スリープスキャンモード

スキャン間にデバイスをスリープモードに移行させることでかなりの電力を節約できます。ただし、スリープ時間を非常に高い値に増加すると、応答時間が遅くなります。デザインで消費電力と応答時間の両方が重要である場合、連続スキャンとスリープスキャン両方のモードを組み合わせた方法を使用できます。この方法を使うと、デバイスは大部分の時間にセンサーをスキャンしてからスリープ状態になるスリープスキャンモードにあります。センサーに触れると、デバイスはセンサーが常にスキャンされる連続スキャンモードに遷移します。デバイスは特定の期間の間、連続スキャンモードのままになります。この期間中にセンサーに触れない場合、デバイスはスリープスキャンモードに戻ります。

7.3.2 長いバックグラウンドループ

バックグラウンドループが CAPSENSE™ 以外のアプリケーションに使用する実行コードが長すぎると、応答時間が遅くなることもあります。Figure 72 に示すように、ループごとに CAPSENSE™ スキャンを複数回実行できます。

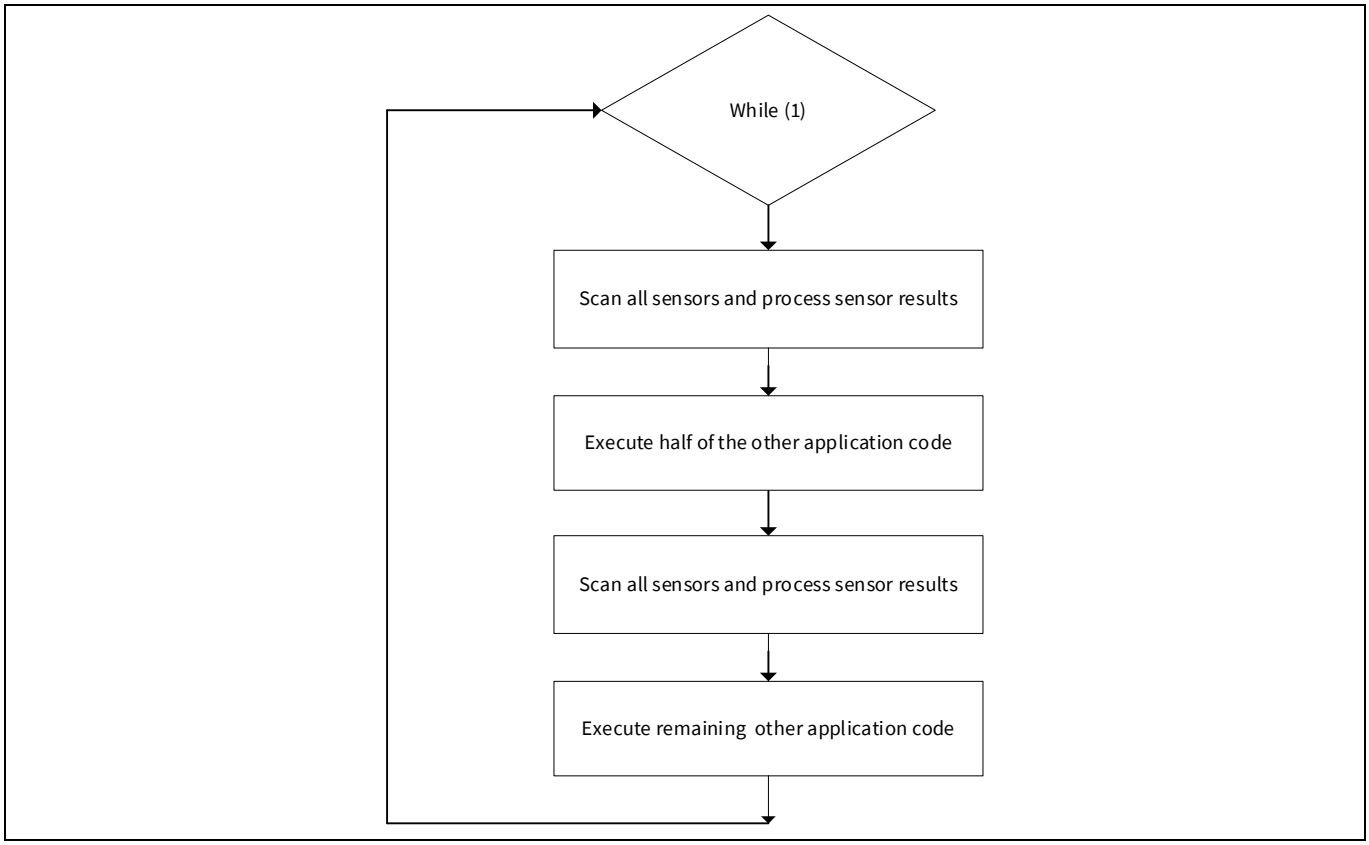

Figure 72　応答時間を改善するループ内の複数回のスキャン

7.3.3　デバウンス

デバウンスは、システムが誤った指タッチを検知することを防ぎますが、Figure 73 に示すように指検知を遅延させます。

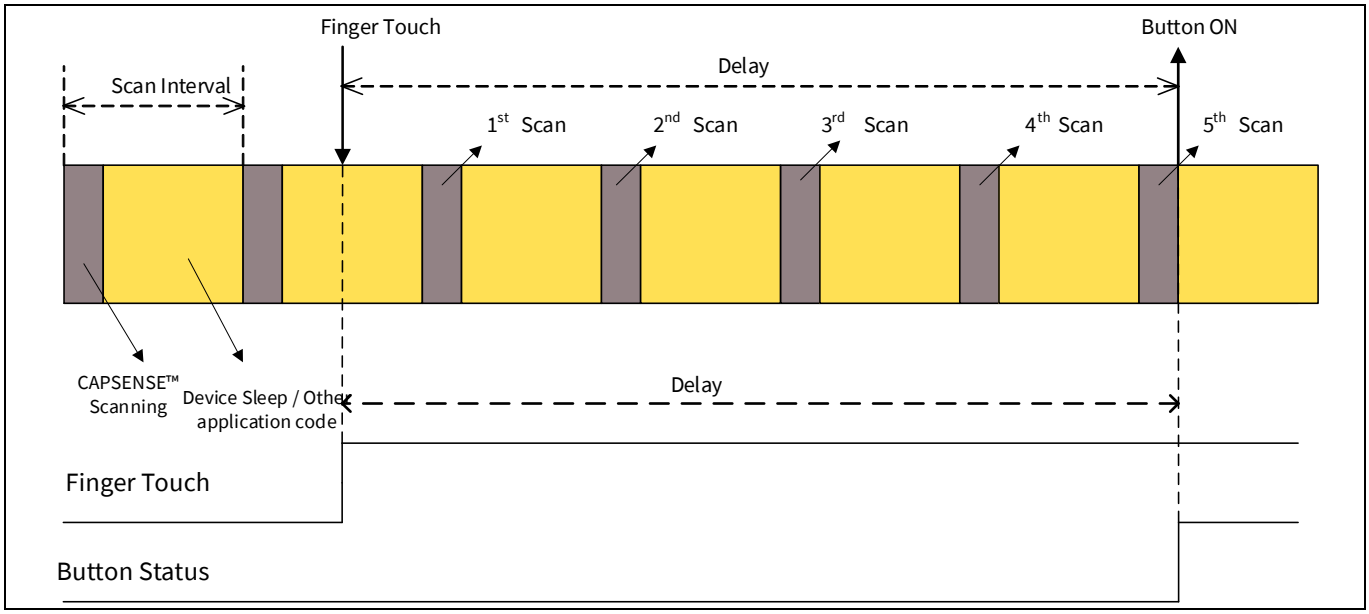

Figure 73　スキャン間隔および検知遅延、デバウンスカウント = 5

デバウンスに起因する最大遅延は以下のとおりです。

$$Max\ Delay = Debounce \times Scan\ Interval$$

Equation 11

スリープモードを使用する場合、デバウンスを 1 に設定してください。スリープモードは応答時間を長くするため、効果的に信号をデバウンスします。これが適切な方法でない場合は以下のプログラムコードに示すように指タッチを検知するとセンサーを複数回スキャンできます。

```
if (CapSense_CheckIsWidgetActive(CapSense_BUTTON0__BTN))
{
    /* Scan the sensor multiple times */
    for (i=0; i < DEBOUNCE ; i++)
    {
        if ( (CapSense_IsBusy() == 0 ) && CapSense_CheckIsWidgetActive(CapSense_BUTTON0__BTN) )
        {
            CapSense_ScanEnabledWidgets();
        }
    }
    /* After number of scans equal to debounce count is complete, check for sensor active.
    * If it is still active, then sensor can be considered as ON.
    */
    if (CapSense_CheckIsWidgetActive(CapSense_BUTTON0__BTN))
    {
        /*  Process the code which is based on sensor active */
    }
}
```

Figure 74　デバウンスプログラムコード

7.3.4　フィルタによる遅延

フィルタをかけることも指タッチの検知を遅延させます。IIR16 フィルタの使用の影響は次のとおりです。IIR16 フィルタは新しいデータの 1/16 および古いデータの 15/16 を追加して、新しいフィルタサンプルを計算することに注意してください。

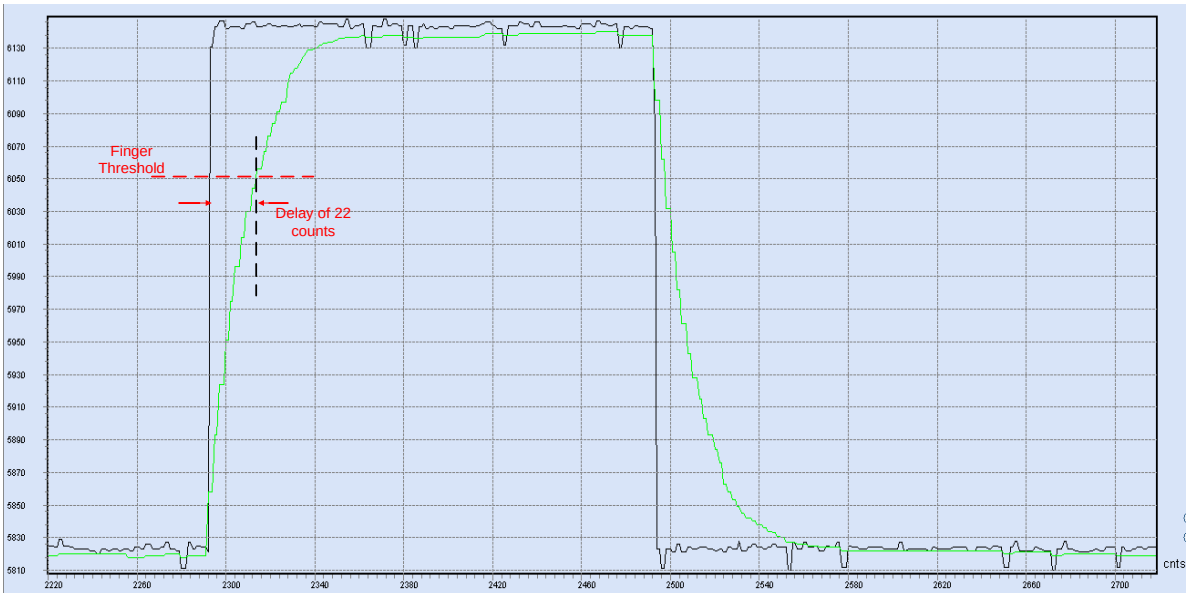

Figure 75　指タッチでのフィルタによる遅延

指によるステップ変更 = 320 カウント

指閾値 = 指のピーク応答の 75% = 240 カウント

フィルタにより発生する遅延の計算は以下のとおりです。

$$T_n = a \times r^{n-1}$$

Equation 12

ここで、

a = 1 (ステップ変更または指のピーク応答)

Tn = 0.25 (ステップ変更の 75%は閾値)

r = 1/16 (フィルタは IIR16)

n = 22

つまり、指タッチは 22 回のスキャン後にのみ検知されます。フィルタによる遅延を最小限にするために、適切なフィルタを選択してください。例えば、IIR4 フィルタは 7 回のスキャンのみの遅延を発生させます。

IIR16 フィルタを使用する必要がある場合、デバウンス で説明したように、複数回のスキャンを使用してください。ただし、センサーはオン/オフ状態にかかわらずループごとに複数回スキャンする必要があります。

7.3.5 割込み優先順位

ノンブロッキングアーキテクチャ で説明したように、Capsense_CSD コンポーネントのコードアーキテクチャは元々ノンブロッキングです。つまり、ハードウェアが CAPSENSE™ をスキャンしている間、CPU はその他のアプリケーションコードを実行できます。ハードウェアがスキャンを完了すると、CPU への割込みを発生します。CAPSENSE™ 割込みの優先順位を変更するために、<プロジェクト名>.cydwr ファイルを開いて [Interrupts] タブをクリックします。

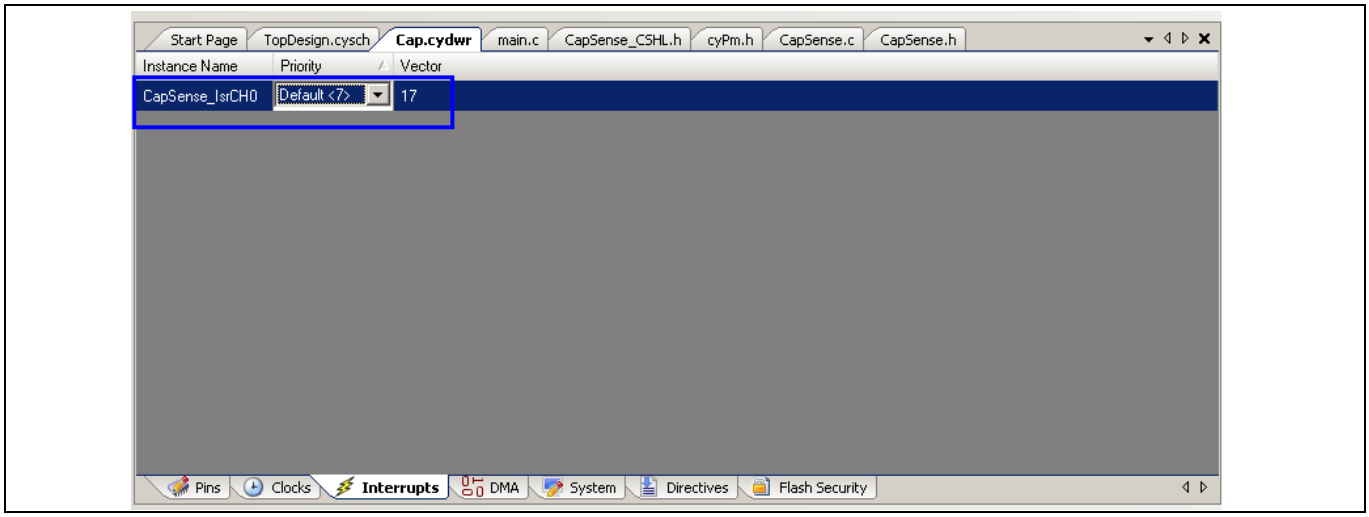

Figure 76 割込み優先順位の変更

通信プロトコルなど優先順位がより高い割込みがある場合、CAPSENSE™ 割込みの優先順位を低くできます。優先順位の高い割込みの処理中に CAPSENSE™ 割込みがアサートされる場合、CAPSENSE™ は割込みが遅れて処理されます。この遅延によりデータ破損は起こりませんが、次のセンサーのスキャンを遅らせ、総スキャン時間を増加させます。

7.4 ピン割り当て

7.4.1 オペアンプ出力ピン

オペアンプ出力は P0[0]、P0[1]、P3[6]、P3[7]ピンに直接接続されます。これらのピンを CAPSENSE™ に使用した場合、オペアンプは使用できなくなります。これらのピンはオペアンプ出力に直接接続しているため、C_P が高くなります。そのため、これら 4 つのピンを CAPSENSE™ に使用しないでください。使用する必要がある場合は、スライダーやタッチパッドに使用せず、ボタンに使用してください。また、C_P を低くするためにピンの配線を短くする必要があります。

7.4.2 2 チャネルデザインのピン割り当て

2 チャネルデザインでは、特定のチャネルに属するピンがチップの同じ側にあるようにします。2 チャネルデザインの詳細については、チャネル数 を参照してください。

7.4.3 C_{MOD} ピン割り当て

最も効果的なアナログルーティングを実現するには、以下のピンを C_{MOD} に使用してください。

- 左側: P2 [0]、P2 [4]、P6 [0]、P6 [4]、P15 [4]
- 右側: P1 [0]、P1[4]、P5 [0]、P5 [4]

7.5 プリント基板レイアウトのガイドライン

詳細なプリント基板のレイアウトガイドラインは CAPSENSE™ 入門 を参照してください。

8 用語集

AMUXBUS

入出力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。

SmartSense 自動チューニング

設計フェーズの後で最適な性能のために、センシングパラメーターを自動設定し、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。

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基準点を提供します。

ボタンまたはボタンウィジェット

関連したセンサーを持って、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有りまたはタッチ無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。差分が負数であるか、またはノイズ閾値未満である場合、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ースソリューション。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インフィニオンの設定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 ファミリーを含んでいます。

重心または重心位置

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。

8 用語集

CSD

CAPSENSE™ シグマデルタ (CSD) は、静電容量センシングアプリケーション用に自己容量を測定するインフィニオンの特許取得済み方法です。

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。

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排除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で (指閾値+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

プリント基板、ITO、または FPCB 上のパッドまたは層などの導電材料です。電極は CAPSENSE™ デバイス上のポートピンに接続し、CAPSENSE™ センサーとして使用されるか、または CAPSENSE™ 機能に関する特定の信号を駆動するために使用されます。

指閾値

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 (指閾値 + ヒステリシス) を上回る場合はオンとして報告され、差分カウントが (指閾値 – ヒステリシス) を下回る場合はオフとして報告されます。

連結センサー

複数のセンサーを連動して、単一の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、電力消費量を減少するために用いられます。

システムが低消費電力モードにある場合の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に、これらの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。ユーザーがいずれかのセンサーに触った場合、システムはアクティブモードに移行し、すべてのセンサーを個別にスキャンしてアクティブになったセンサーを検出します。

PSOC™ はファームウェアでセンサー連結をサポートしており、複数のセンサーは同時に AMUXBUS に接続してスキャンできます。

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よりサポートされます。

8 用語集

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 を設計するとき、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、ラインの幅および 2 本のライン間の間隔は充填率を指定します。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閾値と一緒に使用されるパラメーターです。指閾値 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 動作用の 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閾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 つ以上のセンサーを含んで、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。

8 用語集

変調器クロック

センサーのスキャンの間に 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 (前処理および後処理時間を除く) は ((2^N – 1)/変調器クロック周波数) (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。

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。

負のノイズ閾値

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閾値です。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閾値を超えない (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) が負のノイズ閾値未満である) 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とき、センサーの近くに配置される金属の物体を除去するとき、耐液性のある CAPSENSE™ 製品の水分を除去するとき、および他の急激な環境変化があるときです。

ノイズ (CAPSENSE™ ノイズ)

センサーがオフ状態 (タッチなし) 時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。

ノイズ閾値

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) の差がノイズ閾値を超える場合、おそらく有効な信号を示します。差分がノイズ閾値に下回る場合、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接続されます。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 の配線、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の物体の存在を検知できるセンサーです。

8 用語集

ラジアルスライダー

指の物理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 回の連続スキャン間の時間です。

スキャン分解能

CSD ブロックによって生成される raw カウントの分解能 (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必要な時間です。

自己容量

回路のグラ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、単位を (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ドコンデンサ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

センサーがシステム故障によって誤ったタッチ状態の報告から防ぎ、または金属物体がセンサーの近くに連続的に存在する場合の設定。

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閾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化されていると、ベースラインは差分カウントがノイズ閾値を下回った場合にのみ更新されます。

センサーの連結

連結センサー を参照してください。

8 用語集

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。

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なり、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™ の動作に必要な様々なハードウェアおよびソフトウェアまたは閾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基準電圧ブロックであり、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、リニアスライダー、ラジアルスライダー、マトリックスボタンおよびタッチパッドはサポートされているウィジェットです。

9 参考資料

データシート

PSOC™ 3 および PSOC™ 5LP デバイスファミリーのデータシートは www.infineon.com から入手できます。

- PSOC™ 3 データシート
- PSOC™ 5LP データシート

Technical reference manual (テクニカル リファレンス マニュアル)

次のテクニカル リファレンス マニュアルでは、トップレベル アーキテクチャ図、レジスタのまとめ、タイミング図など PSOC™ 3 および PSOC™ 5LP アーキテクチャに関する情報にすばやく、簡単にアクセスできます。

- PSOC™ 3 リファレンスマニュアル
- PSOC™ 5LP リファレンスマニュアル

開発キット

PSOC™ 3 および PSOC™ 5 開発キット

- CY8CKIT-030 PSOC™ 3 開発キット
- CY8CKIT-050B PSOC™ 5LP 開発キット

MiniProg3

- MiniProg3 は、PSOC™ 3 および PSOC™ 5LP 製品のプログラミングとデバッグに使用されます。また、チューナー GUI の I^2C-USB 通信ブリッジとしても使用されます。

PSOC™ Programmer

- PSOC™ Programmer は、PSOC™ デバイスをプログラミングするための柔軟性のある統合プログラミングアプリケーションです。PSOC™ Designer および PSOC™ Creator を併用して、PSOC™ デバイスに様々なデザインをプログラムできます。
- PSOC™ Programmer は API を備えたハードウェア層を提供し、プログラマとブリッジデバイスを使用して、特定のアプリケーションをデザインします。PSOC™ Programmer ハードウェア層は、COM ガイド文書のほか、C#、C、Perl、および Python の各言語でサンプルコードとしても説明されています。

PSOC™ Creator

- インフィニオンは、独自の統合設計環境である PSOC™ Creator を提供しています。PSOC™ Designer を使用すると、ハードウェアの設定とファームウェアの開発を単一のツールで行えます。

サンプルコード

インフィニオンは、設計を速やかに完了するために大量のサンプルコードを提供しています。PSOC™ Creator を起動し、Start Page で 'find example project' リンクをクリックしてください。CAPSENSE™ に精通するには、以下の 2 つのサンプル プロジェクトを推奨します。

- CapSense_CSD_Design
- CapSense_CSD_With Tuner

9 参考資料

デザインサポート

インフィニオンには、お客様の CAPSENSE™ ソリューションの成功を確実にするさまざまなデザインサポートチャネルがあります。

- 知識ベース記事 (KBA) – 製品ファミリー別の技術情報記事を閲覧したり、CAPSENSE™ についての様々なトピックを検索できます。
- CAPSENSE™ アプリケーションノート – 本書で提供された情報の上に築かれた、幅広いアプリケーションノートを参照します。
- 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習します。
- インフィニオン開発者コミュニティ – インフィニオン技術コミュニティとつながり、情報を交換します。
- 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。
- 技術サポート – 世界一流の技術サポートがオンラインで利用可能。

改訂履歴

版数	発行日	変更内容
**	2012-06-13	これは英語版 001-75400 Rev. **を翻訳した日本語版 001-80490 Rev. **です。
*A	2015-06-30	これは英語版 001-75400 Rev. *B を翻訳した日本語版 001-80490 Rev. *A です。
*B	2020-01-08	これは英語版 001-75400 Rev. *E を翻訳した日本語版 001-80490 Rev. *B です。
*C	2025-12-12	これは英語版 001-75400 Rev. *F を翻訳した日本語版 001-80490 Rev. *C です。

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

Edition 2025-12-12

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2025 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

001-80490 Rev. *C

重要: 注意事項

本文書の全ての日本語情報は、英文で提供されている情報の参考訳です。英文版と和文版に齟齬がある場合は常に英文版の内容が優先します。

製品 (サンプルも含みます。また、ハードウェア、ソフトウェアまたはその両方で構成される製品も含みます。以下同じです。) の販売、提供または引渡しは、インフィニオンテクノロジーズAGおよびその関連会社 (以下まとめて「インフィニオン」といいます)、お客様とインフィニオンによって締結された基本契約その他の書面により合意された条件、そのようなの条件がない場合はインフィニオンの適用ある販売条件に従います。お客様の一般的な取引条件またはインフィニオンの販売条件と乖離した条件は、インフィニオンが書面で明示的に同意した場合に限り、インフィニオンを拘束するものとします。
疑義を避けるため、インフィニオンは、第三者の権利を侵害していないことの保証、および特定の使用/目的に対する適合性または商品性の保証などの黙示的保証を負いません。
インフィニオンは、サンプル、アプリケーション、もしくはお客様による製品の具体的な使 用方法、または本書面に記載された例や標準値に関するいかなる情報についても責任いません。
本書面に記載されているデータは、技術的資格を有する熟練したお客様のご担当者をのみを対象としています。お客様は、意図された用途およびお客様の特定用途に対する製品の適合性を評価し、意図された用途および顧客の特定用途において本書面に含まれるすべての関連技術データを検証する責任を負います。お客様は、意図されたアプリケーションの機能性および安全性を適切に設計、プログラミング、テストし、その使用に関連する法令を遵守してください。
インフィニオンの明示的な承認がない限り、製品は、製品の故障またはその使用による結果が人身傷害につながると合理的に予想されるいかなる用途にも使用しないでください。ただし、上記は、インフィニオンが明示的に設計し、販売した使用分野でお客様が製品を使用することを妨げるものではありません。
インフィニオンは、適用される法律、例えばドイツ著作権法 (UrhG) 第44b条に従って、そのコンテンツを商業的なテキストおよびデータのマイニング (TDM) のために使用する権利を明示的に留保します。
製品にセキュリティ機能が含まれている場合:
いかなるコンピューティング・デバイスも絶対的に安全であることはあり得ないため、製品にセキュリティ対策が施されているにもかかわらず、インフィニオンは、製品に侵入、データの盗難もしくは損失、またはその他の侵害 (「セキュリティ侵害」といいます) がないことを保証せず、インフィニオンは、セキュリティ侵害に起因するいかなる責任も負わないものとします。
本書面がソフトウェアを含んでいるまたはソフトウェアに言及している場合：
ソフトウェアは、米国、ドイツ、および世界各国の知的財産権法および条約に基づきインフィニオンが所有しています。インフィニオンはすべての権利を留保し、無断複写、複製および転写を禁じます。お客様はソフトウェアに同梱されたソフトウェア使用許諾契約に従ってのみソフトウェアを使用できます。
ソフトウェア使用許諾契約が適用されない場合、インフィニオンはここに、ソフトウェアの知的所有権に基づき、(a) ソースコード形式で提供されたソフトウェアについて、インフィニオンのハードウェア製品と共に使用する目的のためにのみソフトウェアを修正および複製するための、そして(b)インフィニオンのハードウェア製品上での使用目的に限り、ソフトウェアをバイナリコード形式でエンドユーザーに外部配布するための、個人的、非排他的、譲渡不能なライセンス (サブライセンス不可) をお客様に付与します。それ以外のソフトウェアの使用、複製、変更、翻訳またはコンパイルは禁止されています。製品、技術、納品条件、および価格に関する詳細については、最寄りのインフィニオンオフィスに連絡するか、https://www.infineon.comにアクセスしてください。