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

本書について

適用範囲と目的

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、CAPSENSE™入門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。

対象者

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。

目次

本書について . . . 1

目次 . . . 1

1 はじめに . . . 9

1.1 概要 . . . 9

1.2 CAPSENSE™の特長 . . . 9

1.3 PSOC™ 4 および PSOC™ 6 MCU CAPSENSE™ Plus の特長 . . . 10

1.4 CAPSENSE™の設計フロー . . . 11

2 CAPSENSE™技術 . . . 14

2.1 CAPSENSE™の基礎 . . . 14

2.1.1 自己容量センシング . . . 15

2.1.2 相互容量センシング . . . 17

2.2 静電容量タッチ センシング方式 . . . 18

2.2.1 CAPSENSE™シグマ デルタ (CSD) . . . 19

2.2.2 CAPSENSE™クロスポイント (CSX) . . . 19

2.3 信号対ノイズ比 (SNR) . . . 21

2.4 CAPSENSE™のウィジェット . . . 22

2.4.1 ボタン (0 次元) . . . 22

2.4.2 スライダー (1 次元) . . . 26

2.4.3 タッチパッド/トラックパッド (2 次元) . . . 27

2.4.4 **近接 (3 次元)** . . . 28

目次

2.5 耐液性 . . . 29
2.5.1 自己容量センシングの耐液性 . . . 30
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 . . . 30
2.5.1.2 被駆動シールド信号およびシールド電極 . . . 34
2.5.1.3 ガード センサー . . . 35
2.5.2 相互容量センシングの耐液性 . . . 36
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 . . . 36
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 . . . 36
2.5.3 耐液性性能への液体属性の影響 . . . 38

3 PSOC™ 4 および PSOC™ 6 MCU CAPSENSE™ . . . 39
3.1 PSOC™ 4 および PSOC™ 6 の CAPSENSE™世代 . . . 39
3.2 Capacitive Sigma-Delta (CSD) センシング方式 (第 3 および第 4 世代) . . . 41
3.2.1 GPIO セルの静電容量-電流変換器 . . . 42
3.2.2 IDAC ソース モード . . . 43
3.2.3 IDAC シンク モード . . . 45
3.2.4 CAPSENSE™クロック ジェネレーター . . . 46
3.2.4.1 センス クロック . . . 46
3.2.4.2 変調器クロック . . . 47
3.2.5 シグマ デルタ変換器 . . . 47
3.2.6 アナログ マルチプレクサ (AMUX) . . . 49
3.2.7 CAPSENSE™ CSD シールド . . . 49
3.3 CAPSENSE™ CSX センシング方式 (第 3 および第 4 世代) . . . 49
3.4 CAPSENSE™ CSD-RM センシング方式 (第 5 世代および第 5 世代低電力) . . . 52
3.4.1 GPIO セルの容量から電荷へのコンバーター . . . 54
3.4.2 コンデンサ DAC (CDAC) . . . 57
3.4.3 CAPSENSE™クロック ジェネレーター . . . 57
3.4.3.1 センス クロック . . . 57
3.4.3.2 変調器クロック . . . 57
3.4.4 レシオメトリックセンシングテクノロジー . . . 58
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) . . . 60
3.4.6 CAPSENSE™ CSDRM シールド . . . 60
3.4.6.1 アクティブシールド . . . 60
3.4.6.2 パッシブシールド . . . 61
3.5 CAPSENSE™ CSX-RM センシング方式 (第 5 世代および第 5 世代低消費電力) . . . 62
3.5.1 レシオメトリックセンシングテクノロジー . . . 64
3.6 第 5 世代および第 5 世代低消費電力の利点 . . . 66
3.6.1 自律スキャン . . . 66
3.6.2 マルチチャネルの使用 . . . 66
3.6.3 CIC2 . . . 66

4 CAPSENSE™設計および開発ツール . . . 68

目次

4.1 PSOC™ Creator . . . 68
4.1.1 CAPSENSE™コンポーネント . . . 68
4.1.2 CapSense_ADC コンポーネント . . . 69
4.1.3 チューナー GUI . . . 69
4.1.4 サンプル プロジェクト . . . 70
4.2 ModusToolbox™ . . . 70
4.2.1 CAPSENSE™ ミドルウェア . . . 70
4.2.2 CAPSENSE™設定ツール . . . 71
4.2.3 CSDADC ミドルウェア . . . 71
4.2.4 CSDIDAC ミドルウェア . . . 71
4.2.5 CAPSENSE™チューナー . . . 71
4.2.6 サンプル プロジェクト . . . 72
4.3 ハードウェア キット . . . 72

5 CAPSENSE™性能のチューニング . . . 75
5.1 SmartSense および手動チューニングの選択 . . . 75
5.2 SmartSense . . . 75
5.2.1 概要 . . . 75
5.2.2 SmartSense の全自動チューニング . . . 77
5.2.2.1 チューニングボタンウィジェット . . . 77
5.2.2.2 チューニング スライダー ウィジェット . . . 80
5.2.2.3 チューニング近接ウィジェット . . . 81
5.2.3 SmartSense ハードウェア パラメーター – Only モード . . . 81
5.2.4 初期チューニングの SmartSense . . . 81
5.3 手動チューニング . . . 82
5.3.1 概要 . . . 82
5.3.2 CSD センシング方式 (第 3 および第 4 世代) . . . 83
5.3.2.1 基本 . . . 84
5.3.2.1.1 変換利得および CAPSENSE™信号 . . . 84
5.3.2.1.2 フラット スポット . . . 88
5.3.2.2 CAPSENSE™ハードウェア パラメーターの選択 . . . 89
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 . . . 89
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 . . . 90
5.3.2.2.3 シールド電極の調整 . . . 93
5.3.2.3 CAPSENSE™ソフトウェア パラメーターの選択 . . . 95
5.3.2.3.1 ベースライン . . . 96
5.3.2.3.2 ベースライン更新アルゴリズム . . . 97
5.3.2.3.3 指しきい値 . . . 97
5.3.2.3.4 ヒステリシス . . . 98
5.3.2.3.5 ノイズしきい値 . . . 98
5.3.2.3.6 負のノイズしきい値 . . . 98
5.3.2.3.7 低ベースライン リセット . . . 99

目次

5.3.2.3.8 デバウンス . . . 99
5.3.2.3.9 センサー自動リセット . . . 99
5.3.2.3.10 マルチ周波数スキャン . . . 100
5.3.2.4 ボタン ウィジェット調整 . . . 100
5.3.2.5 スライダー ウィジェット調整 . . . 102
5.3.2.6 タッチパッドウィジェットの調整 . . . 105
5.3.2.7 近接ウィジェット調整 . . . 105
5.3.3 CSX センシング方式 (第 3 および第 4 世代) . . . 105
5.3.3.1 基本 . . . 105
5.3.3.1.1 変換利得および CAPSENSE™信号 . . . 105
5.3.3.2 CAPSENSE™ハードウェア パラメーターの選択 . . . 107
5.3.3.2.1 Tx クロック パラメーター . . . 107
5.3.3.2.2 変調器クロック周波数 . . . 108
5.3.3.2.3 IDAC . . . 108
5.3.3.2.4 サブ変換回数 . . . 109
5.3.3.3 CAPSENSE™ソフトウェア パラメーターの選択 . . . 109
5.3.3.4 ボタン ウィジェット調整 . . . 109
5.3.3.5 タッチパッドウィジェットの調整 . . . 111
5.3.4 CSD-RM センシング方式 (第 5 世代および第 5 世代低消費電力) . . . 111
5.3.4.1 基本 . . . 111
5.3.4.1.1 変換利得および CAPSENSE™信号 . . . 111
5.3.4.1.2 フラット スポット . . . 116
5.3.4.2 CAPSENSE™ハードウェア パラメーターの選択 . . . 117
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 . . . 118
5.3.4.2.2 ハードウェア パラメーターの手動チューニング . . . 119
5.3.4.2.3 シールド電極の調整 . . . 127
5.3.4.2.4 CAPSENSE™ソフトウェア パラメーターの選択 . . . 130
5.3.4.2.5 自律スキャンの設定 . . . 130
5.3.4.2.6 マルチチャネルスキャン . . . 135
5.3.4.2.7 ボタン ウィジェット調整 . . . 137
5.3.4.2.8 スライダー ウィジェット調整 . . . 137
5.3.4.2.9 タッチパッドウィジェットの調整 . . . 137
5.3.4.2.10 近接ウィジェットの例 . . . 137
5.3.4.2.11 低消費電力ウィジェット チューニング . . . 137
5.3.5 CSX-RM センシング方式 (第 5 世代および第 5 世代低消費電力) . . . 137
5.3.5.1 基本 . . . 137
5.3.5.1.1 変換利得および CAPSENSE™信号 . . . 137
5.3.5.2 CAPSENSE™ハードウェア パラメーターの選択 . . . 140
5.3.5.2.1 スキャンモード . . . 140
5.3.5.2.2 センサー接続方法 . . . 140
5.3.5.2.3 変調器クロック周波数 . . . 140

目次

5.3.5.2.4 初期化サブ変換 141
5.3.5.2.5 Tx クロック パラメーター 141
5.3.5.2.6 サブ変換回数 143
5.3.5.2.7 容量性 DAC 143
5.3.5.2.8 補償 CDAC 分周器 145
5.3.5.2.9 自動キャリブレーション 145
5.3.5.2.10 CDAC コードの選択 146
5.3.5.2.11 CDAC ディザ 146
5.3.5.3 CAPSENSE™ソフトウェア パラメーターの選択 146
5.3.5.4 自律スキャンの設定 146
5.3.5.5 マルチチャネルスキャン 146
5.3.5.6 ボタン ウィジェット調整 146
5.3.5.7 タッチパッドウィジェットの調整 147
5.3.6 手動チューニングのトレードオフ 147
5.3.6.1 信頼性 148
5.3.6.2 消費電力および応答時間 148
5.3.7 チューニング デバッグ FAQ 149
5.3.7.1 チューナーがデバイスと通信しない 149
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 149
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 149
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 149
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 150
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 150
5.3.7.7 システムは 85%に校正できない 150
5.3.7.8 スライダーの重心応答が非線形である 151
5.3.7.9 スライダー セグメントは大きな変動のある CP がある 151
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 151
5.3.7.11 SNR が低くなっている 153
5.3.7.12 CSX ボタンの CM が低いことを確認 154

6 CAPSENSE™のジェスチャー 157
6.1 タッチジェスチャーの対応 157
6.2 ジェスチャー グループ 157
6.3 1 本指ジェスチャーの実装 157
6.3.1 ウィジェットの調整 157
6.3.2 事前定義されたジェスチャーの選択 158
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 159
6.3.4 ジェスチャーパラメーターの調整 159
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 160
6.3.4.2 クリック 161
6.3.4.2.1 シングルクリック 162
6.3.4.2.2 ダブルクリック 163

目次

6.3.4.3 スクロール . . . 164
6.3.4.3.1 1 本指スクロール . . . 164
6.3.4.3.2 1 本指の慣性スクロール . . . 165
6.3.4.4 1 本指フリック . . . 165
6.3.5 2 本指のジェスチャーの実装 . . . 165
6.3.6 ジェスチャーの高度なフィルター . . . 166

7 設計上の注意事項 . . . 167
7.1 ファームウェア . . . 167
7.1.1 低消費電力の設計 . . . 169
7.1.2 同期スキャン . . . 171
7.2 センサーの構造 . . . 173
7.3 オーバーレイの選択 . . . 174
7.3.1 オーバーレイの素材 . . . 174
7.3.2 オーバーレイの厚さ . . . 175
7.3.3 オーバーレイの接着剤 . . . 176
7.4 PCB レイアウト ガイドライン . . . 176
7.4.1 センサー CP . . . 176
7.4.2 回路基板の層 . . . 177
7.4.3 ボタン デザイン . . . 177
7.4.3.1 自己容量のボタン デザイン . . . 177
7.4.3.2 相互静電容量ボタンの設計 . . . 178
7.4.3.2.1 フィッシュボーンパターン . . . 178
7.4.3.2.2 任意の形状と寸法のボタンデザイン . . . 179
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 . . . 181
7.4.4 スライダーの設計 . . . 184
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 . . . 185
7.4.4.2 スライダーの端におけるダミー セグメント . . . 190
7.4.4.3 スライダー寸法の決定 . . . 191
7.4.4.4 スライダー セグメント配線の経路 . . . 192
7.4.4.5 LED 付きスライダー デザイン . . . 192
7.4.5 センサーとデバイスの配置 . . . 192
7.4.6 配線の長さと幅 . . . 193
7.4.7 配線の経路 . . . 193
7.4.8 クロストークに対するソリューション . . . 195
7.4.9 ビア . . . 195
7.4.10 グランド面 . . . 196
7.4.10.1 E-pad のないパッケージの使用 . . . 197
7.4.10.2 E-pad のあるパッケージの使用 . . . 199
7.4.10.3 PSOC™ 4 Bluetooth® LE デバイスの使用 . . . 199
7.4.11 電源レイアウトの推奨事項 . . . 201
7.4.12 耐液性のレイアウト ガイドライン . . . 202

目次

7.4.12.1 シールド電極のレイアウトガイドライン 202
7.4.12.2 ガード センサーのレイアウトガイドライン 204
7.4.12.3 グランド リングによる耐液性 205
7.4.13 回路図のルール チェックリスト 205
7.4.13.1 外部コンデンサ接続のピン選択 206
7.4.13.2 センサー ピンの選択 208
7.4.14 レイアウト ルール チェックリスト 210
7.5 CAPSENSE™システムのノイズ 213
7.5.1 Finger Injected ノイズ 213
7.5.1.1 Finger injected ノイズを減らすための推奨事項 213
7.5.2 VDDA ノイズ 214
7.5.2.1 VDDA ノイズを低減するための推奨事項 215
7.5.3 外部ノイズ 215
7.5.3.1 ESD 保護 215
7.5.3.1.1 ESD 放電の防止 216
7.5.3.1.2 リダイレクト 216
7.5.3.1.3 ESD 保護デバイス 217
7.5.3.2 電磁適合性 (EMC) に対する考慮事項 217
7.5.3.2.1 放射干渉およびエミッション 217
7.5.3.2.2 伝導性の RF ノイズ 231
7.6 接地の影響 231
7.6.1 CSX 方式 231
7.6.1.1 CbodyDG>>Cfs 232
7.6.1.2 CbodyDG<<Cfs 233
7.6.2 CSD 方式 234
7.6.2.1 AC/DC 駆動のアプリケーション 234
7.6.2.2 バッテリ駆動のアプリケーション 235

8 CAPSENSE™ Plus 236

9 リソース 241
9.1 ウェブサイト 241
9.2 デバイス データシート 241
9.3 コンポーネント データシート/ミドルウェア ドキュメント 241
9.4 テクニカル リファレンス マニュアル 241
9.5 開発キット 241
9.6 PSOC™ Creator 241
9.7 ModusToolbox™ 241
9.8 アプリケーション ノートおよびデザイン ガイド 242
9.9 設計サポート 242

用語集 243

改訂履歴 249

目次

商標 254
免責事項 255

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 - PSOC™ 4 入門または AN91267 - 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 - PSOC™ Creator での PSOC™ 6 Bluetooth® LE 入門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 の CAPSENSE™テクノロジー MCU
- PSOC™ 4 および PSOC™ 6 MCU 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 PSOC™6 MCU 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 調理器などの 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 (SNR) 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。

- **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

[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。

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ー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。

PSOC™ 4000T デバイスは、第 5 世代の CAPSENSE™ およびマルチセンス技術を搭載した PSOC™ 4 MCU ファミリのメンバーです。**常時オン**の統合センシング技術に基づく超低消費電力のタッチ HMI ソリューション、優れた耐液性を備えた現代的で洗練されたユーザ インターフェース ソリューションを可能にする性能の向上、過酷な環境向けの堅牢で信頼性の高いタッチ HMI ソリューションを提供します。

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

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 - CAPSENSE™入門を参照してください。

1.4 CAPSENSE™の設計フロー

図 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。表 1 に、図 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。

1 はじめに

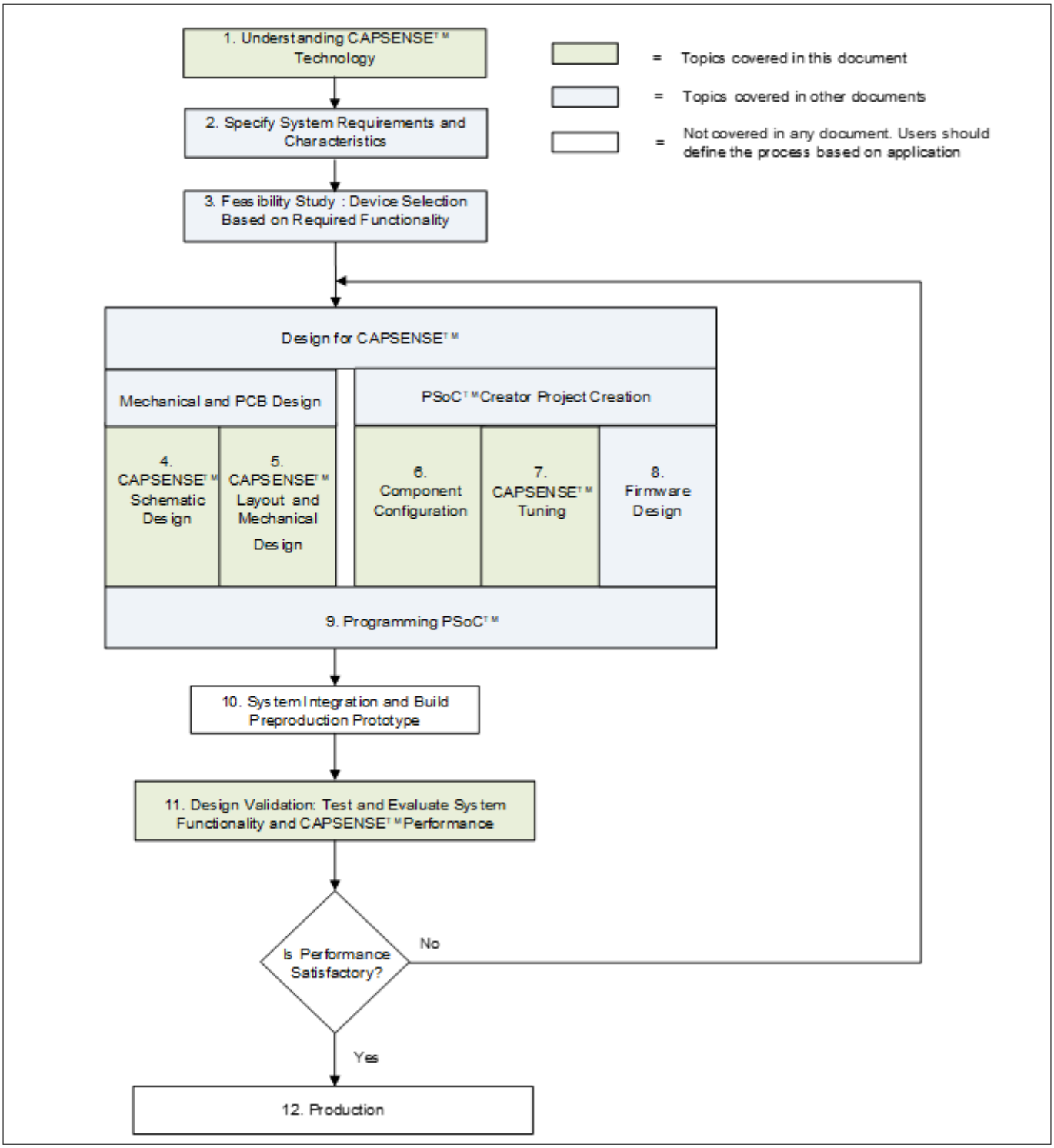


図 1　　CAPSENSE™製品の設計フロー

表 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) CAPSENSE™入門	第 2 章および第 3 章 –

(続く)

表 1 (続き) 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
2.要件定義	CAPSENSE™入門	–
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 – CAPSENSE™入門 AN79953 – PSOC™ 4 入門 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 ドキュメント	–
	CAPSENSE™デザイン ガイド (本書)	第 5 章
7.性能チューニング	PSOC™ 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 ドキュメント	–
	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™の関連ドキュメント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	–	–

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。図 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。

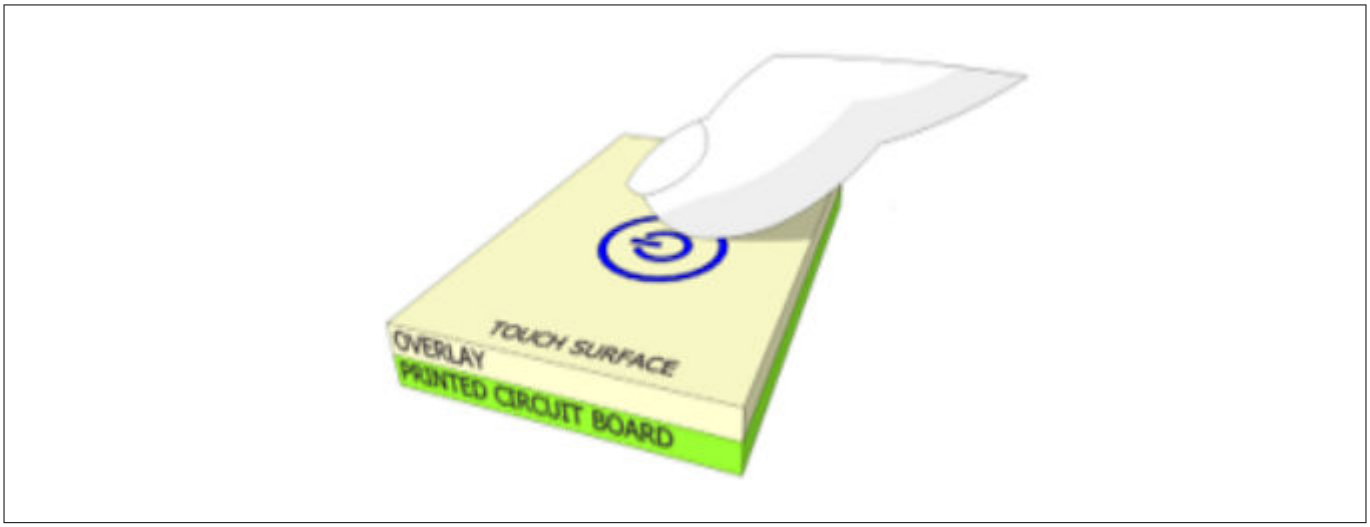


図 2 静電容量タッチ センサー

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。図 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 (詳細については第 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 ピンとグランドの間に接続されます。表 38 に、外部コンデンサの推奨容量値を示します。

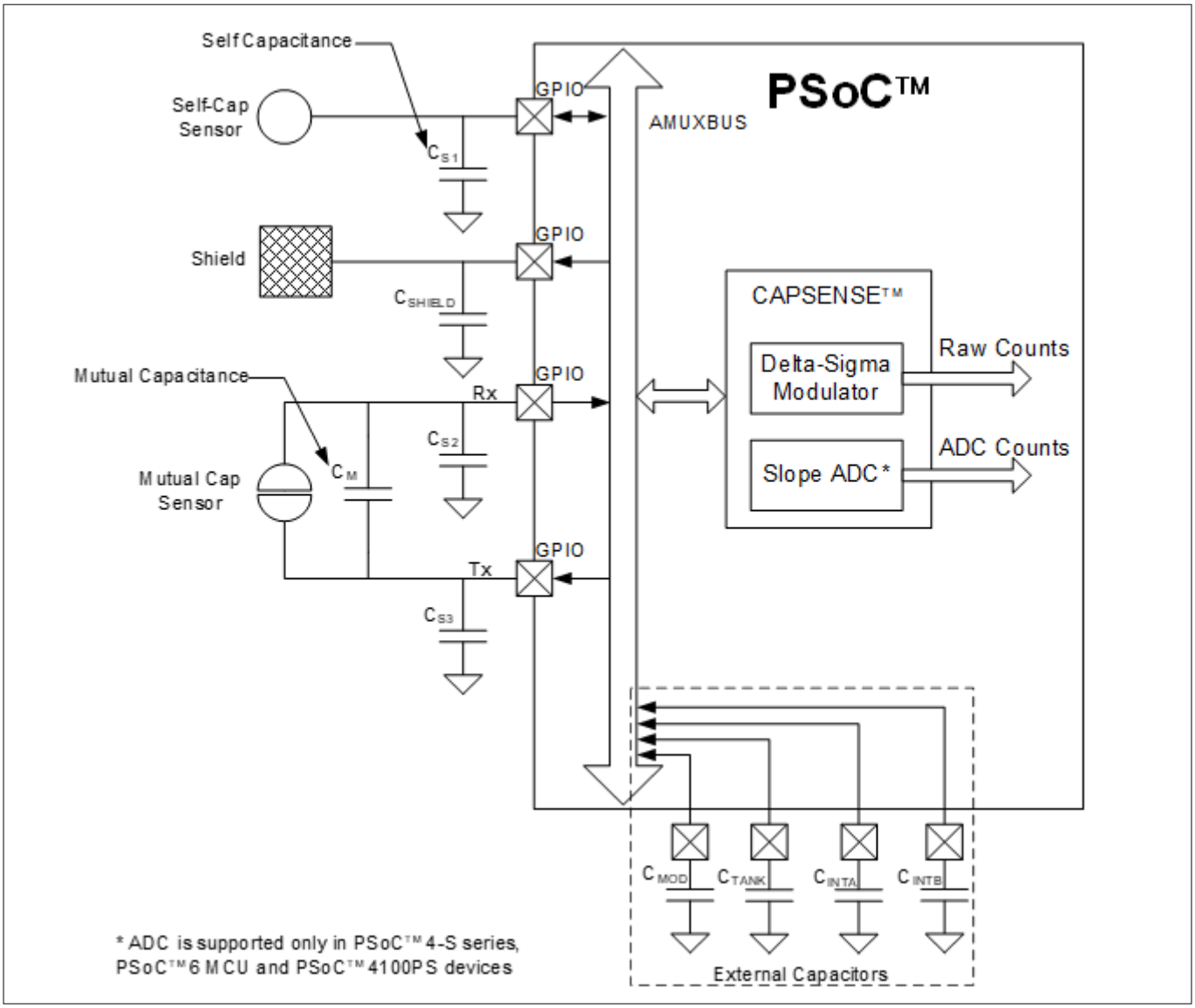


図 3　　PSOC™デバイス, センサー, および外部コンデンサ

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

図 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。図 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。

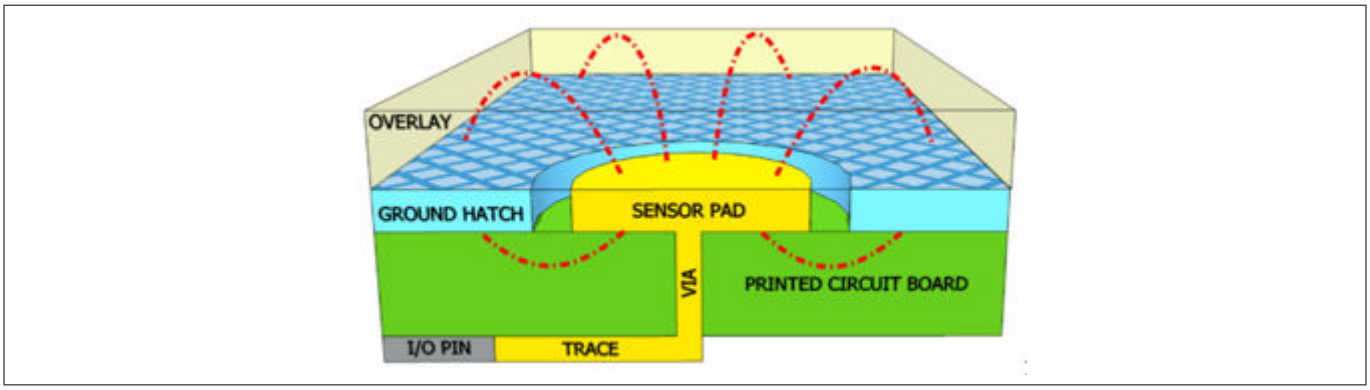


図 4　　寄生容量

図 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。

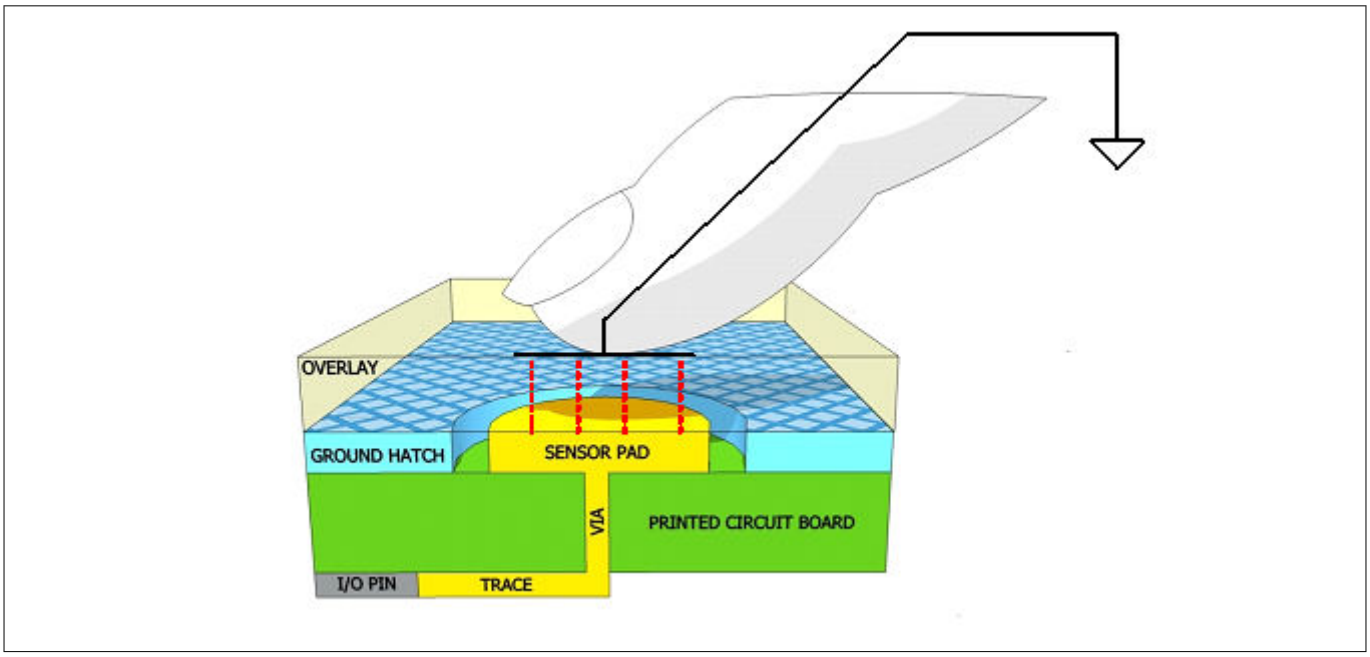


図 5　　指の静電容量

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。

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

式 1　　指の静電容量

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。

$$C_s = C_P + C_F$$

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

タッチしていない場合は、C_S は C_P です。

PSOC™は、容量 CS 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 CP 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 ドキュメントの CSD 仕様を参照してください。

2.1.2　相互容量センシング

図 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、VDDIO[2] (VDDIO は I/O ピンに電源を供給します。) または VDDD[3] (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) に正比例します。

図 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^1_M 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。

2

3　VDDD はデジタル セクションに電源を供給します。

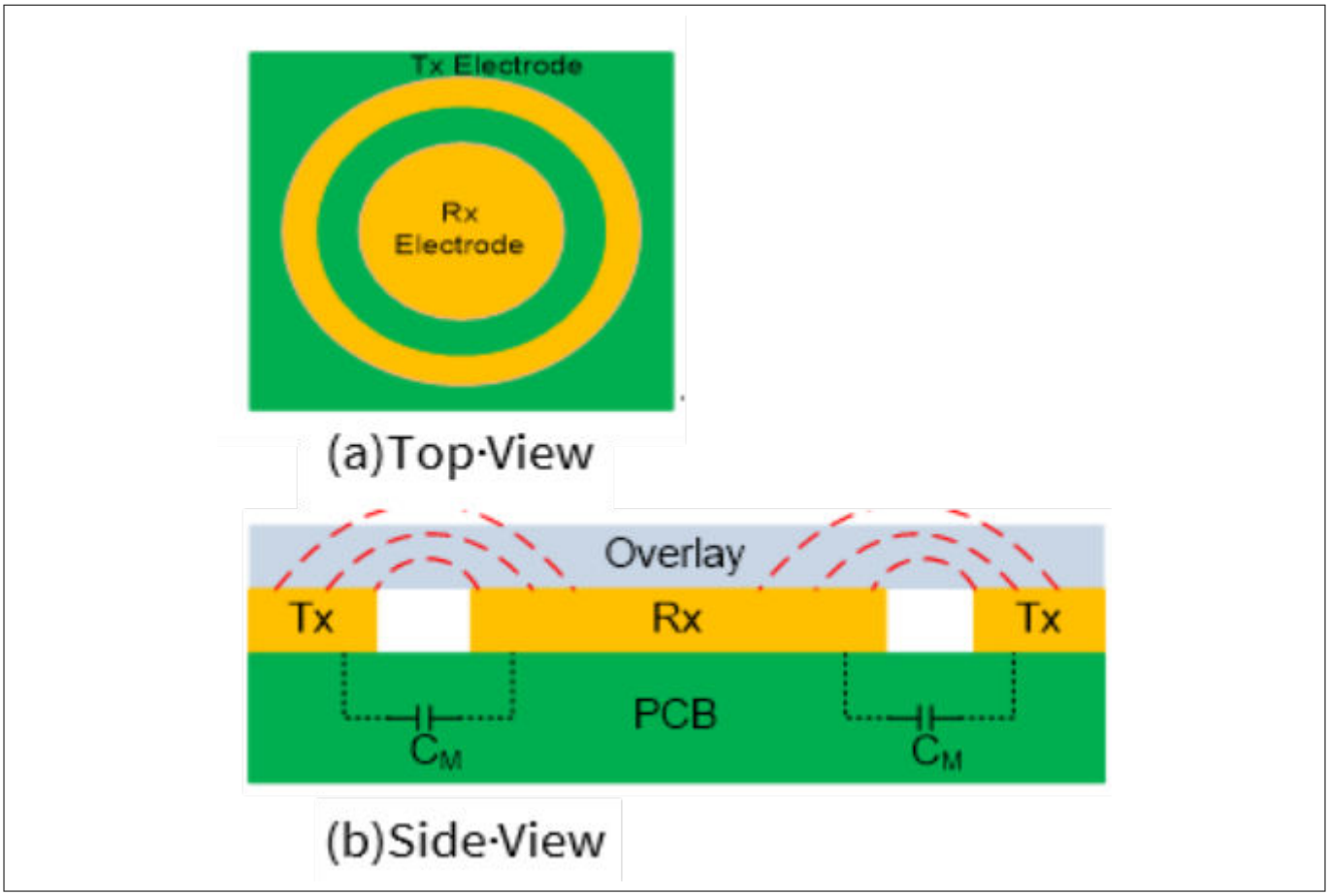


図 6　　相互容量センシング

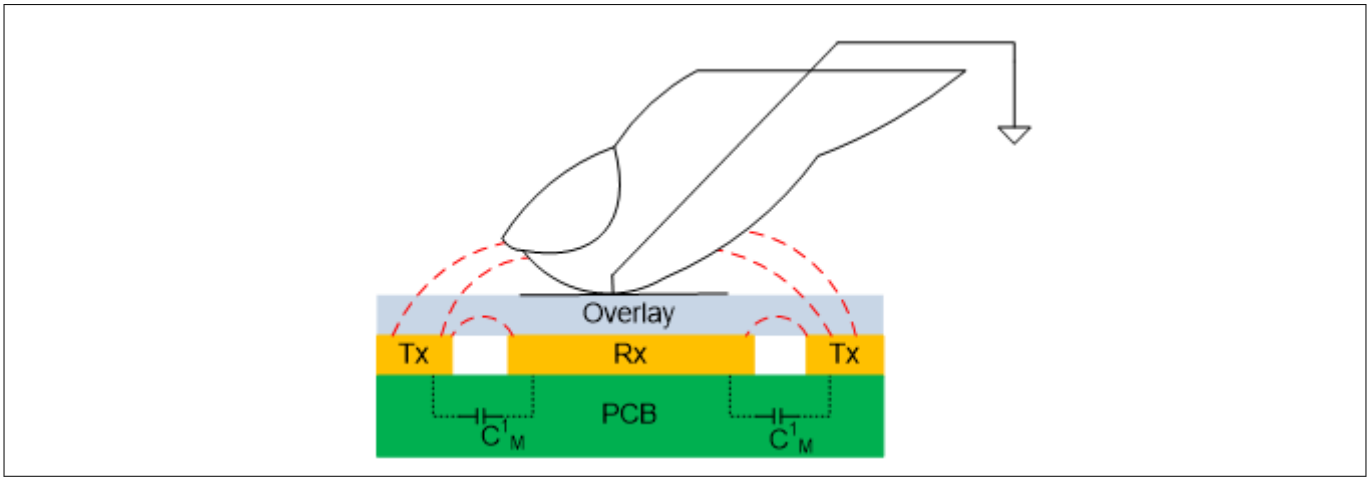


図 7　　指のタッチがあるときの相互容量

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)

図 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。

$$\text{raw count} = G_{CSD}\, C_S$$

式 3 CSD 内の raw カウントとセンサー静電容量の関係

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

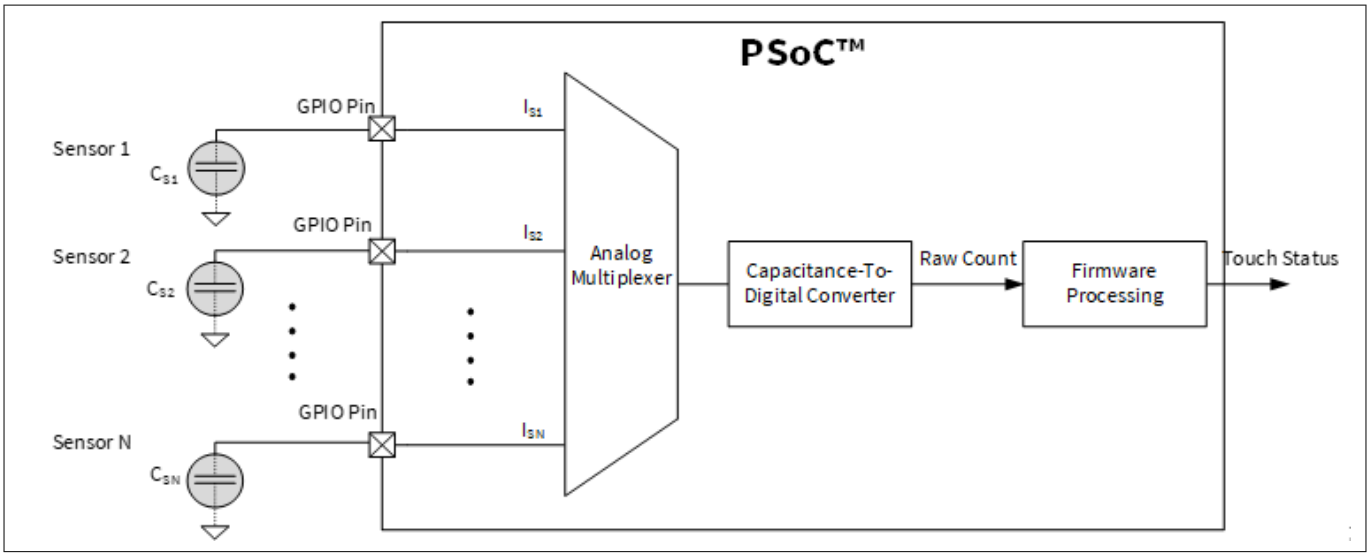


図 8 CSD 方式の概略図

図 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。

2.2.2 CAPSENSE™クロスポイント (CSX)

図 9 は CSX 方式を簡略化したブロック図を示します。

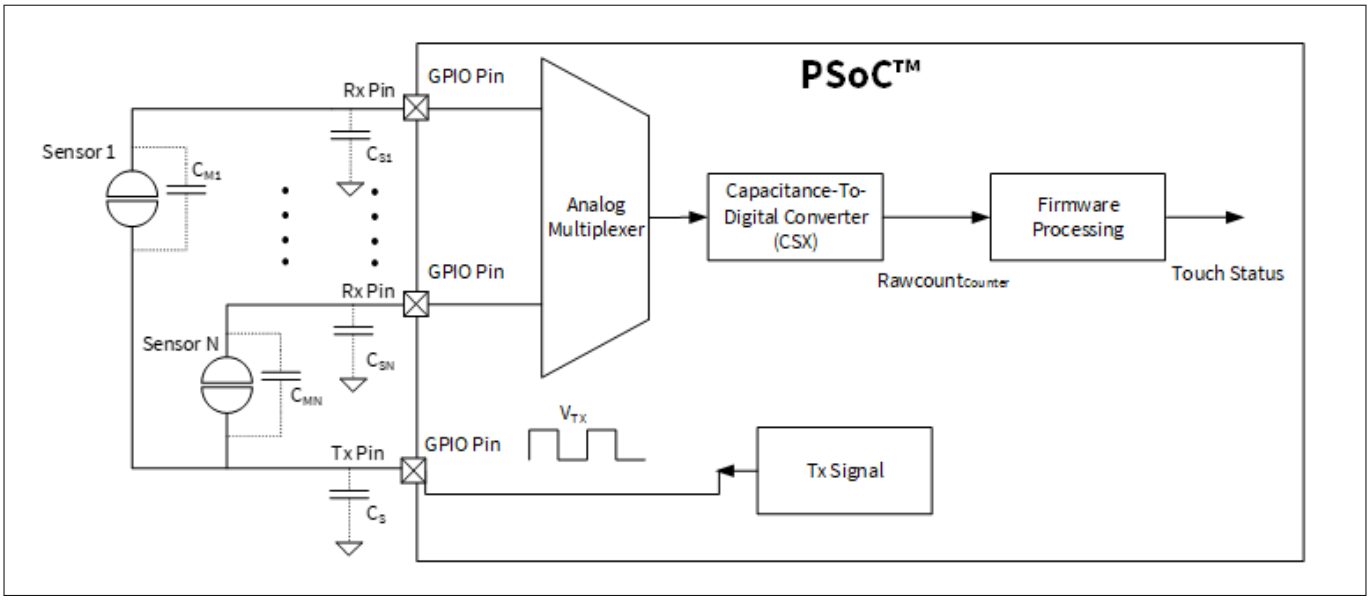


図 9　　CAPSENSE™クロスポイント (CSX) 方式の概略図

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。

電流-デジタル変換器の出力カウント、$Rawcount_{Counter}$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。

$$Rawcount_{Counter} = G_{CSX}\, C_M$$

式 4　　CSX 内の raw カウントとセンサー静電容量の関係

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

図 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 から C^1_M に低下するため (図 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。

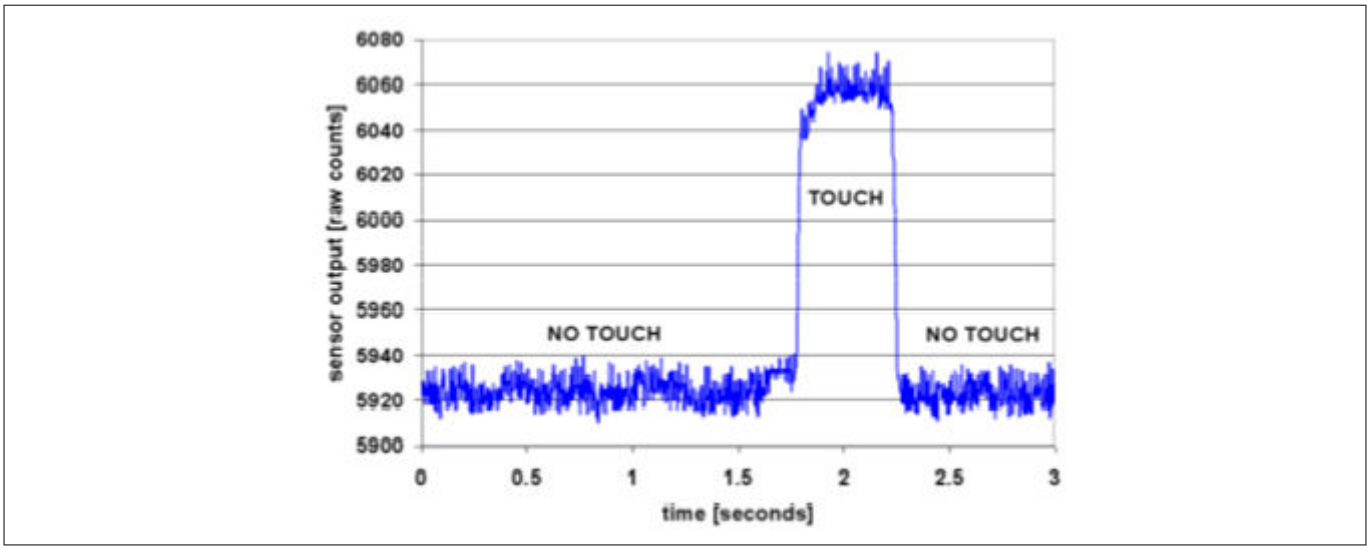


図 10　　raw カウントと時間

PSOC™ 4 および PSOC™ 6 CAPSENSE™ CSD および CSX ブロックの詳細については 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。図 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ーです。

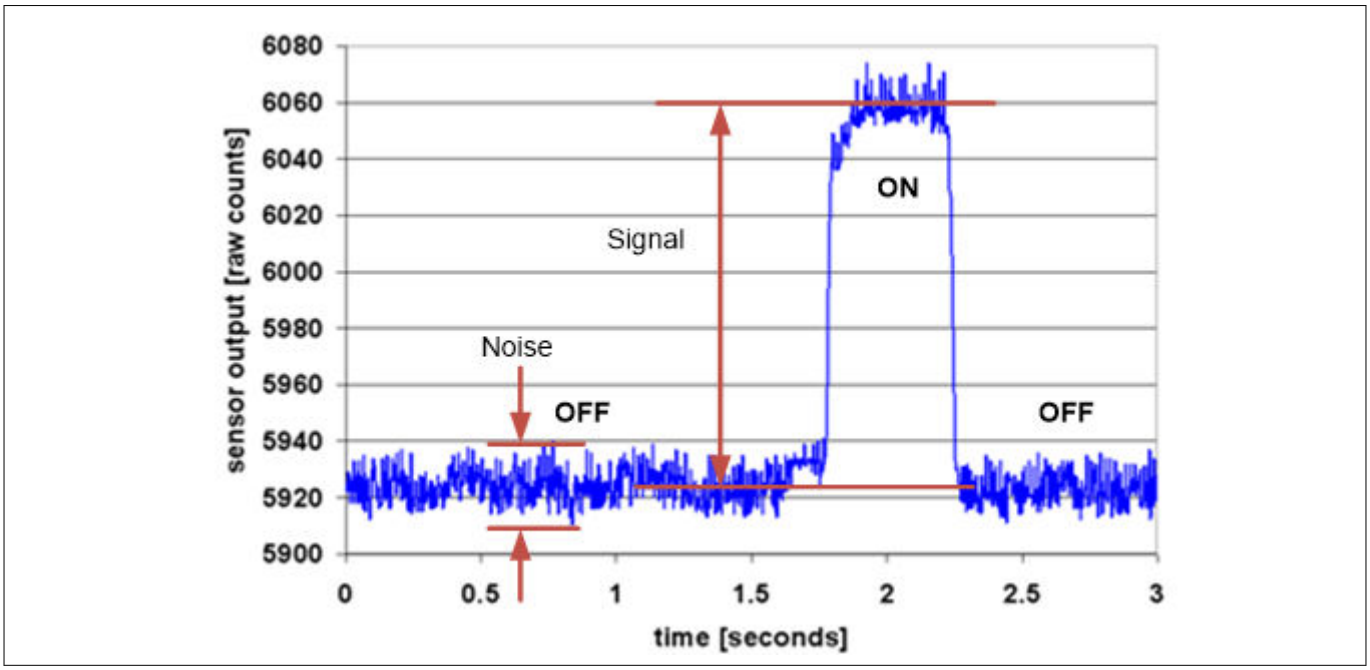


図 11　　SNR

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™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。図 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についてはセンサーの構造を参照してください。

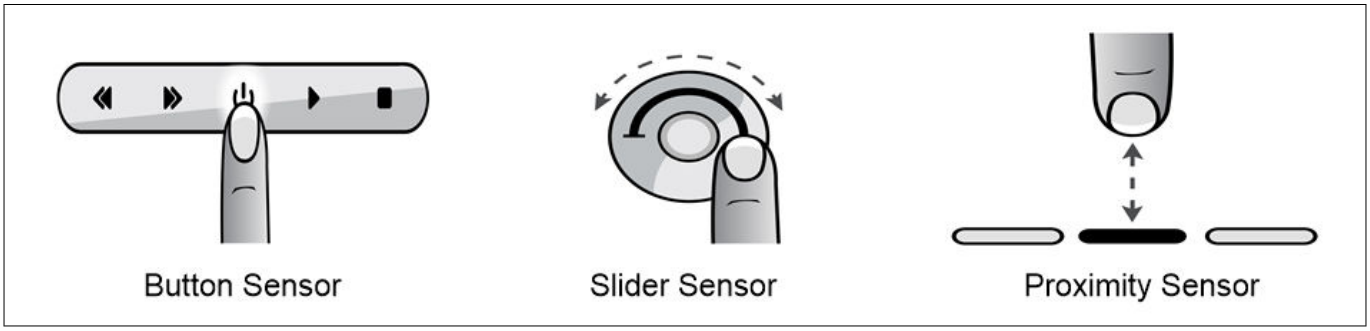


図 12 ウィジェットの種類

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。CAPSENSE™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。

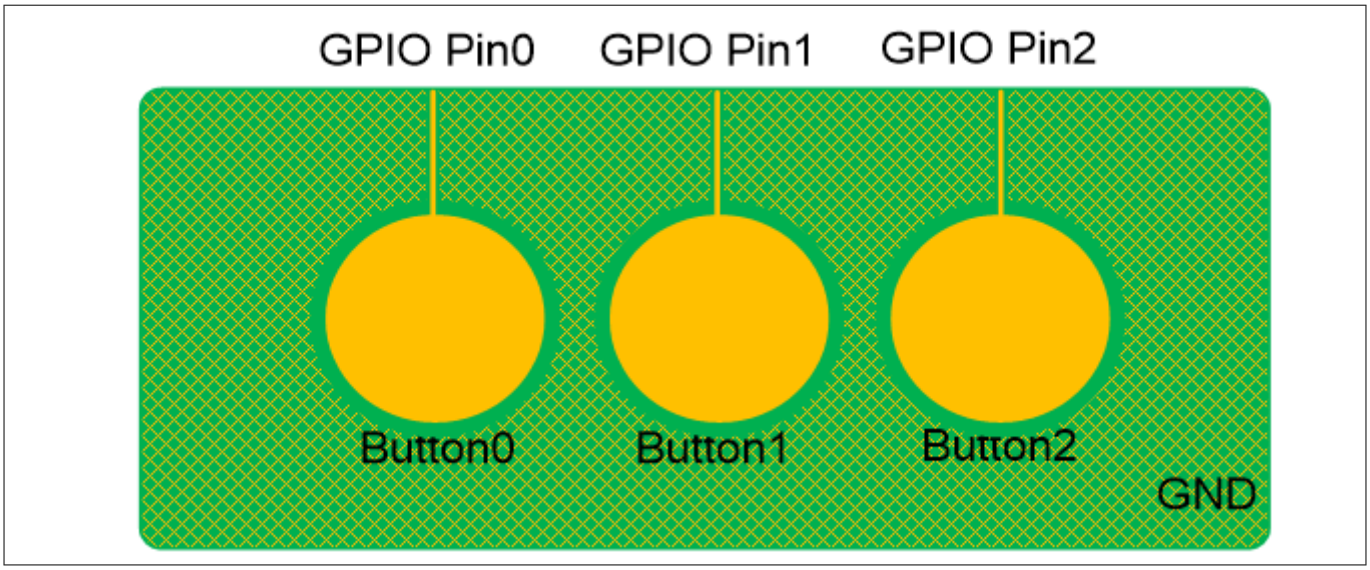


図 13　　簡単な CAPSENSE™ボタン

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。Tx ピンと Rx ピンは複数のボタンで共有でき、各ボタンは 図 14 のように Rx と Tx のユニークな交差点になります。しかし、Rx へのノイズのカップリングを最小にするため、Rx トレースの長さを最小にし、Rx トレースの両側をグランドでシールドすることを推奨します。

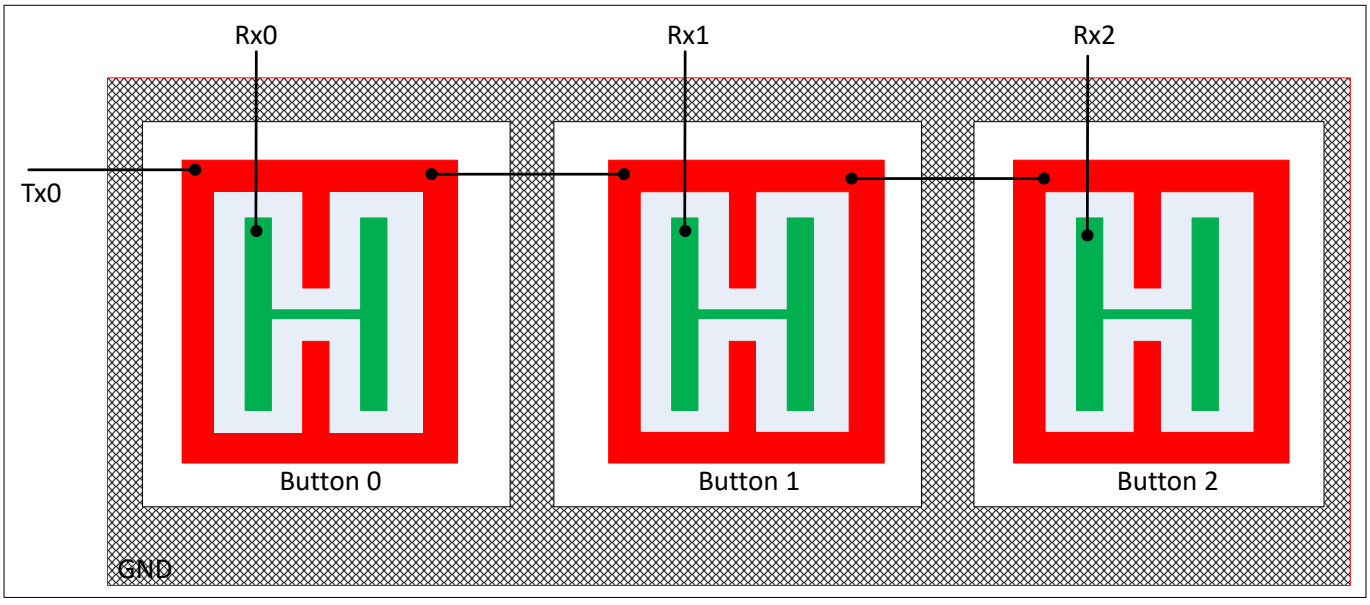


図 14　　相互容量センシング方式での簡単な CAPSENSE™ボタン

図 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™ボタンをマトリッ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、図 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。

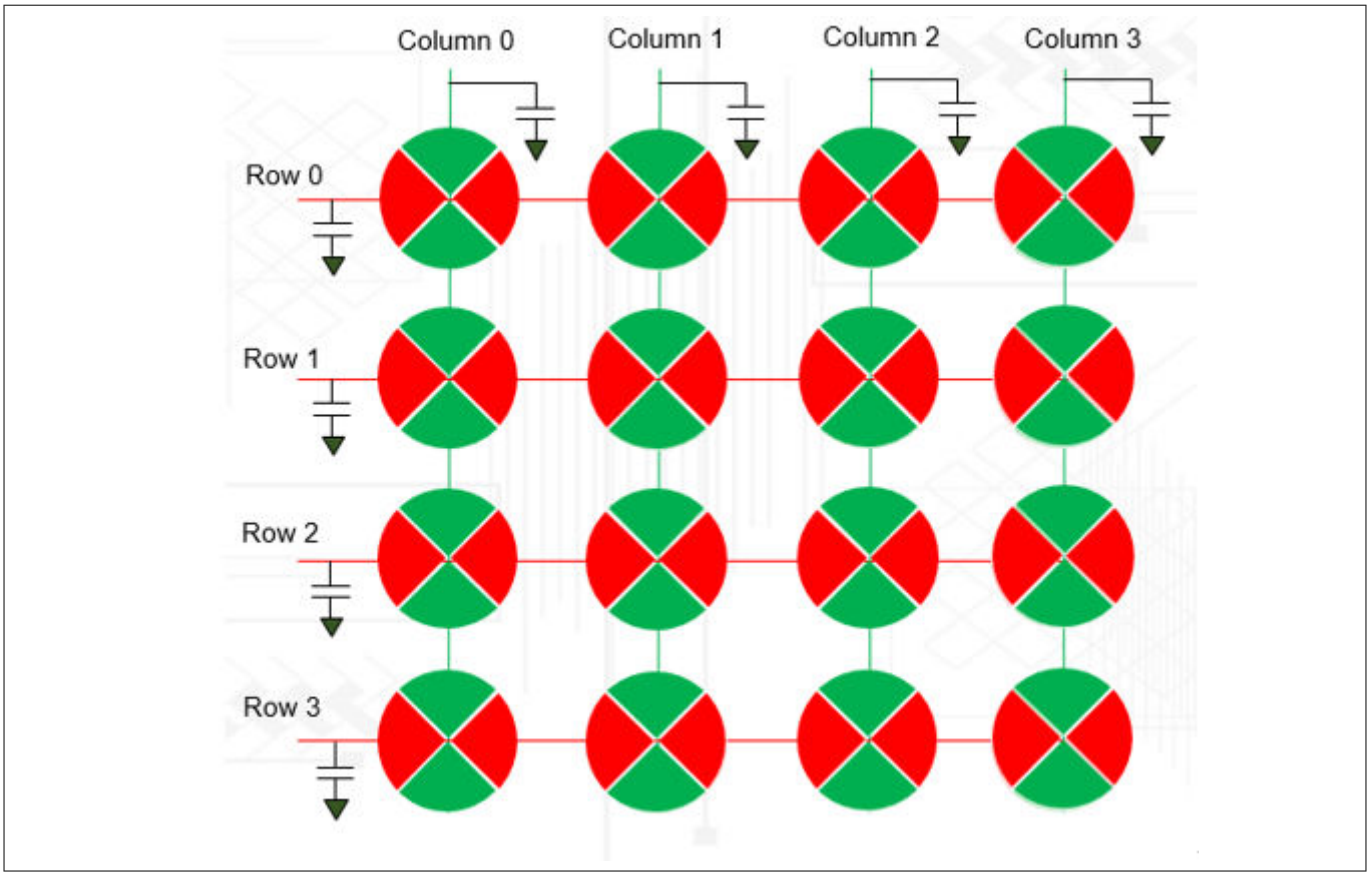


図 15　　CSD を基にしたマトリックス ボタン

マトリックス 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。マトリックス 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。

図 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 ベースのマトリックスボタンは、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、図 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。

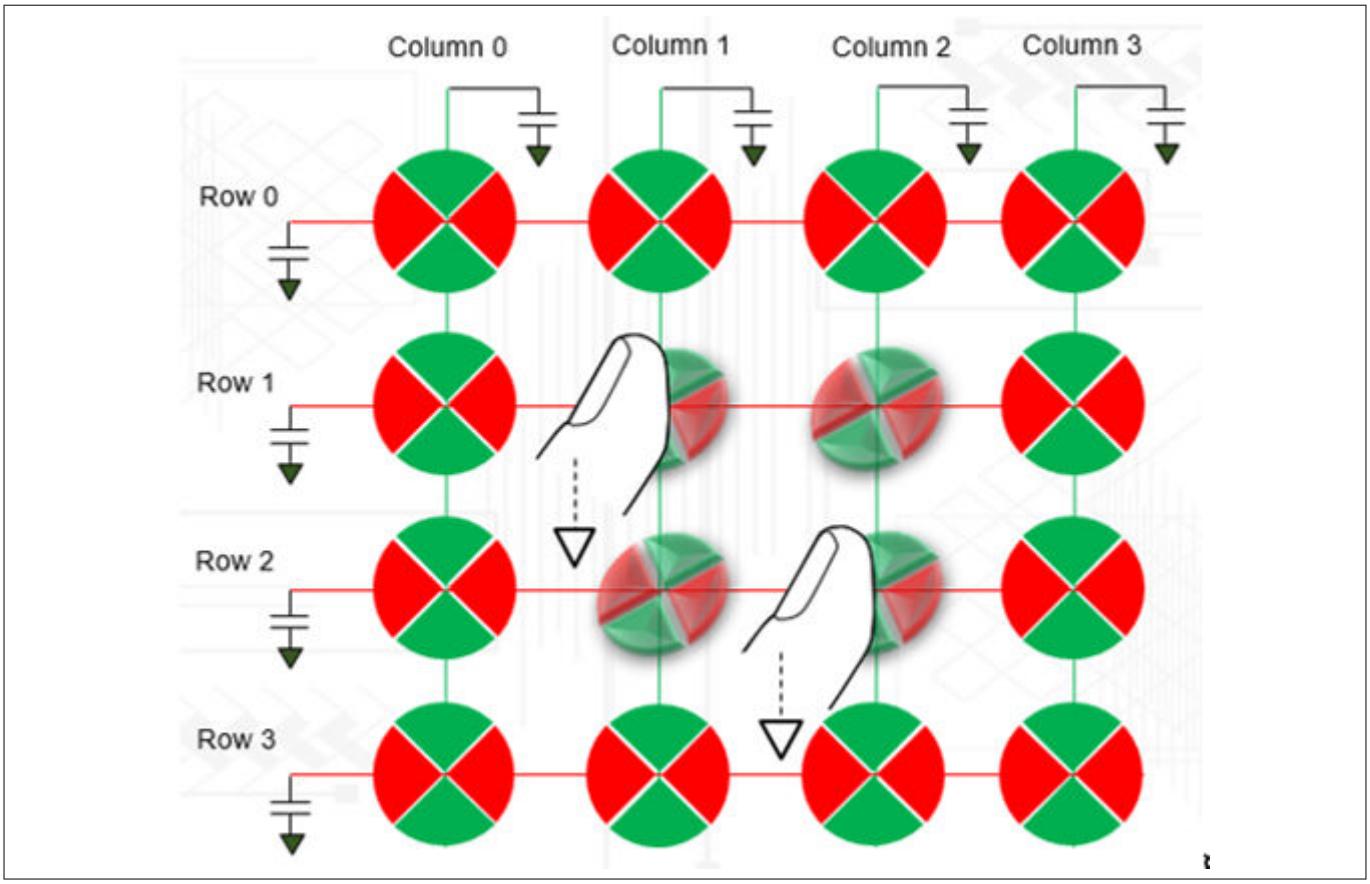


図 16　　CSD を基にしたマトリックス ボタンのゴースト効果

相互静電容量は、ゴーストタッチ現象の影響を受けず、高 C_p 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、図 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。

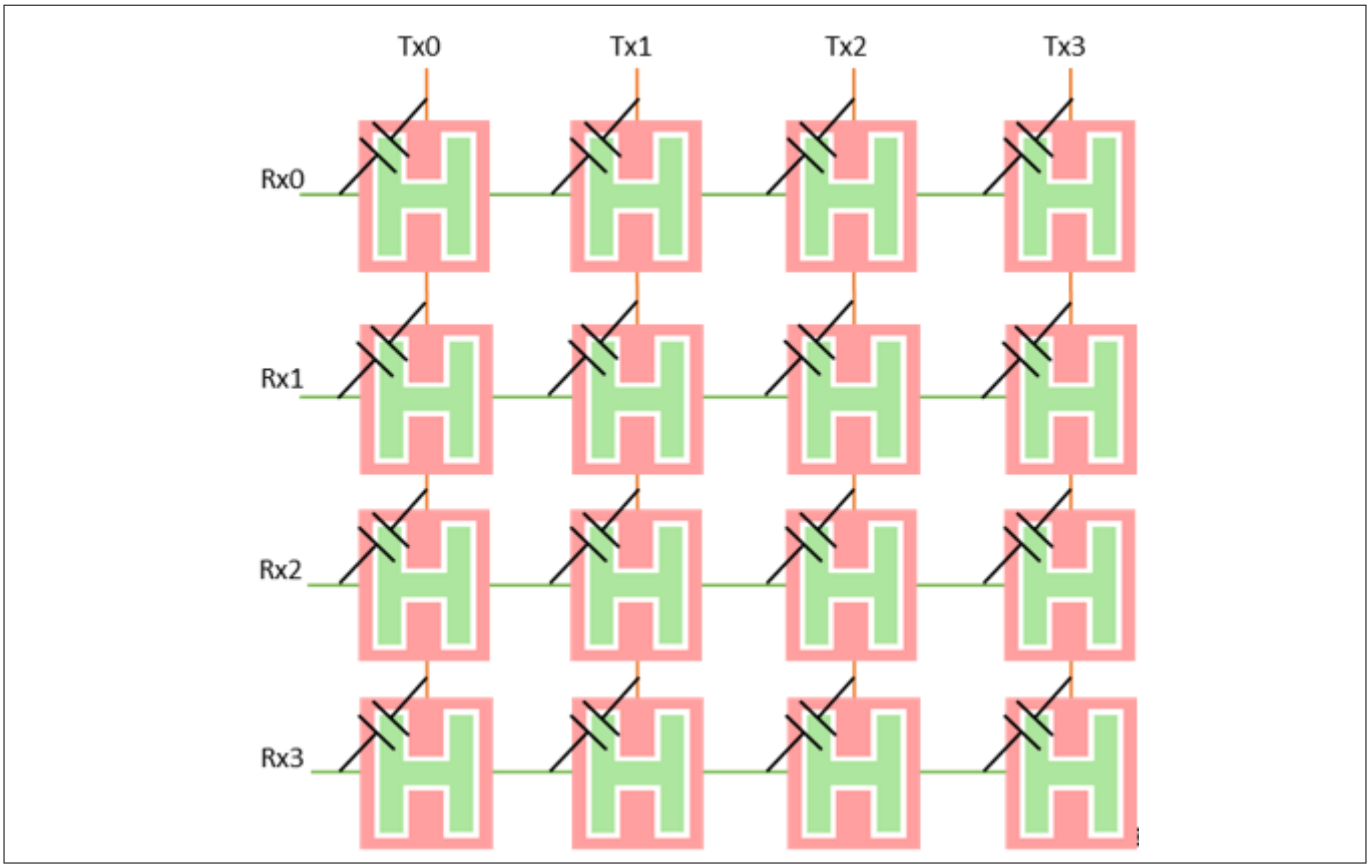


図 17　　CSX を基にしたマトリックス ボタン

注:　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。

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傍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 ヶ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。

図 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。

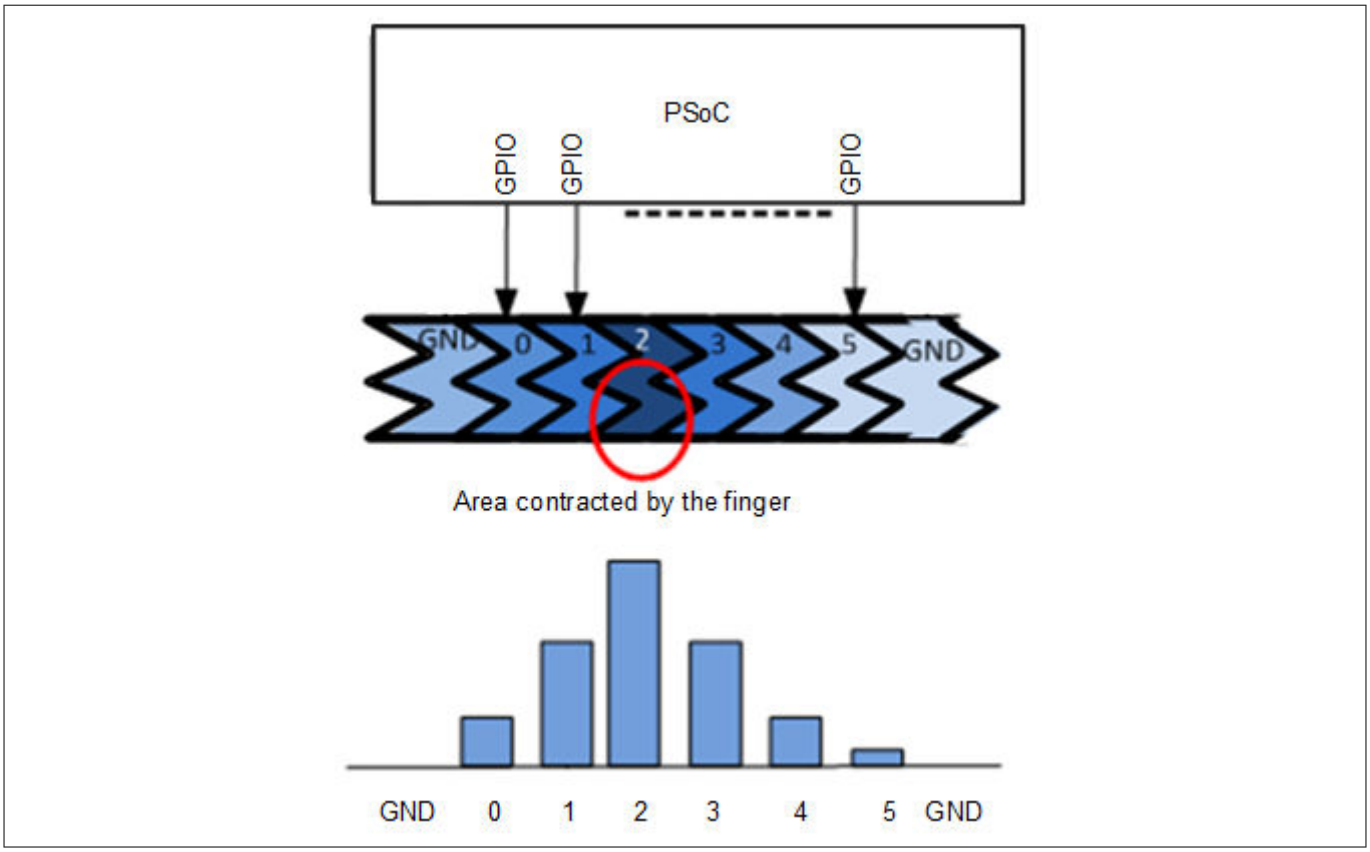


図 18　　リニア スライダー

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。図 19 に、一般的なラジアル スライダーを示します。

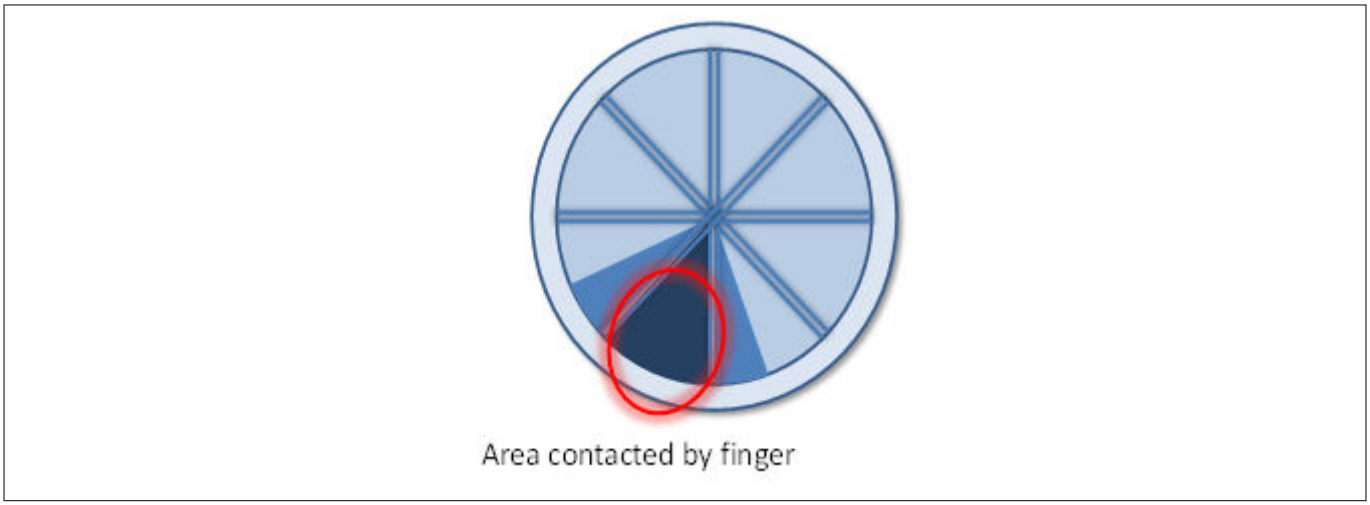


図 19　　ラジアル スライダー

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 の長さと 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 スライダーがあります。図 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。

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。

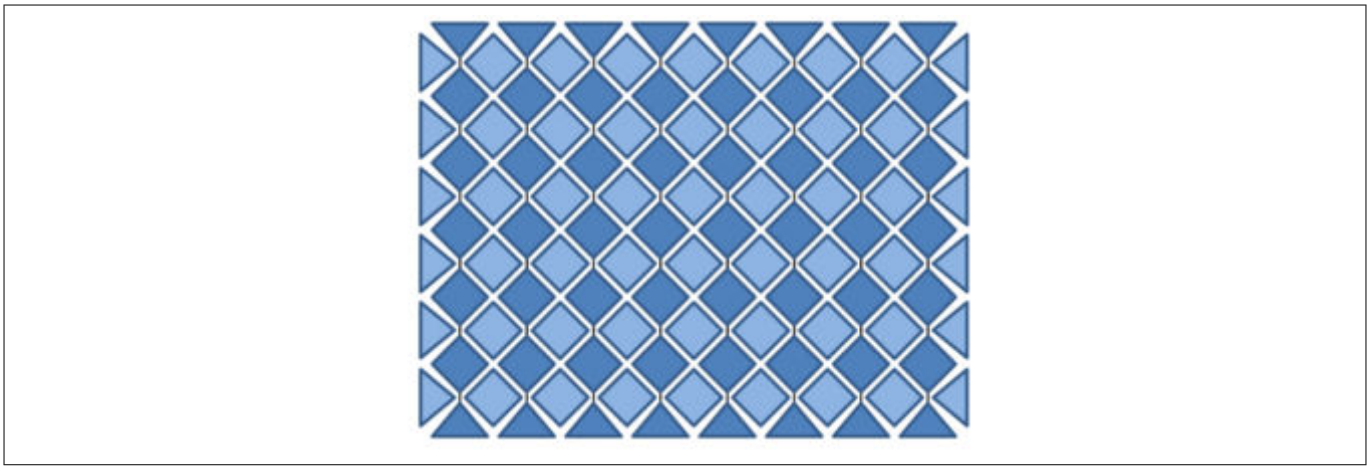


図 20　　タッチパッド センサーの配置

詳細については、AN234185 PSOC™ 4 CAPSENSE™ Touchpad design guide を参照してください。

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の出力は、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、ボタンやスライダー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。図 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。

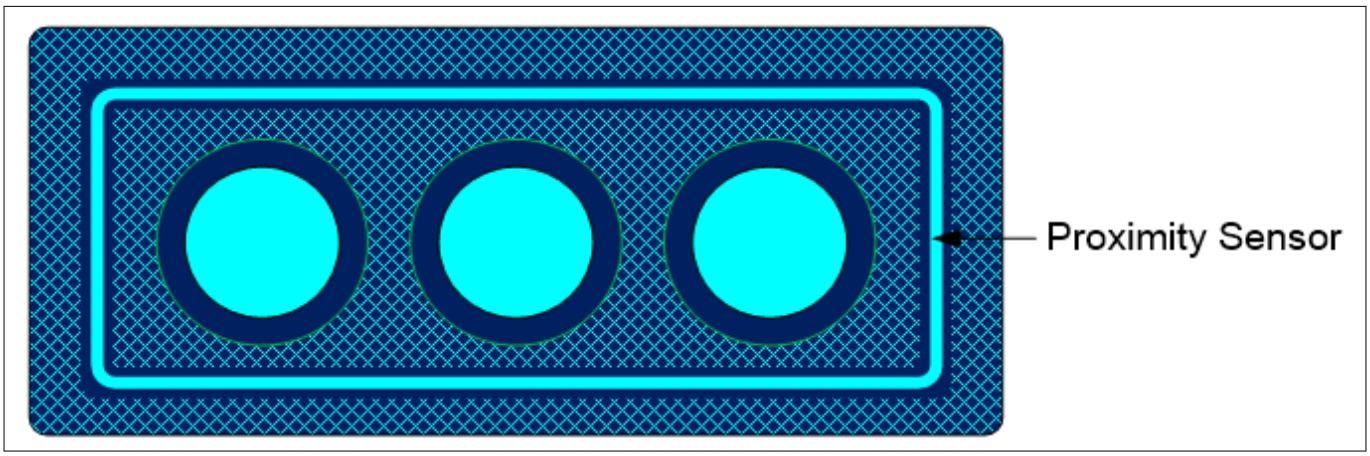


図 21　　近接センサー

他のセンサーを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 ドキュメントを参照してください。

近接センサーの詳細については、AN92239 - CAPSENSE™による近接センシングおよび CAPSENSE™入門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。



図 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、図 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。
- ***注**: 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、図 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は、図 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。

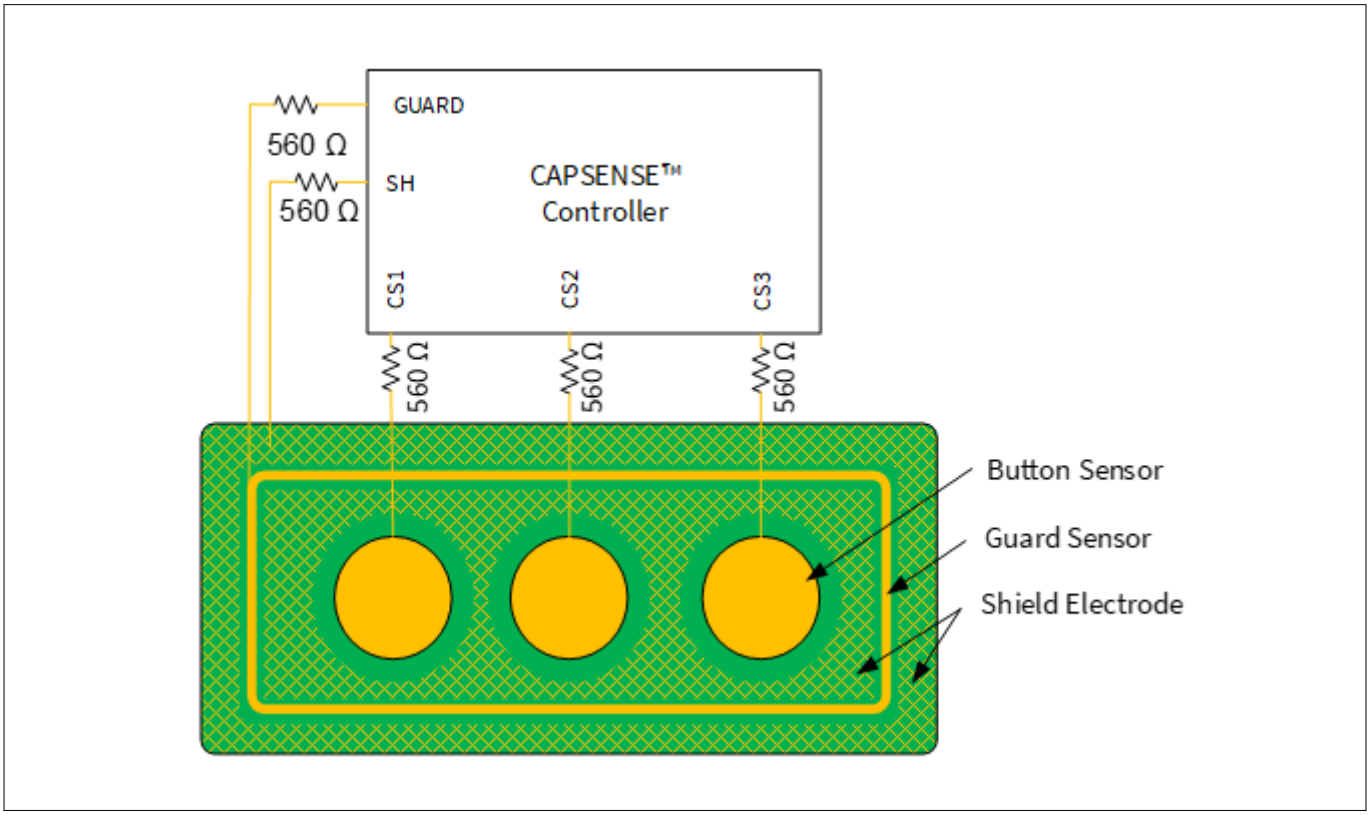


図 23　　CAPSENSE™コントローラーに接続したシールド電極 (SH) およびガード センサー (GUARD)

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、図 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、図 24(b) では C_P と表記されます。

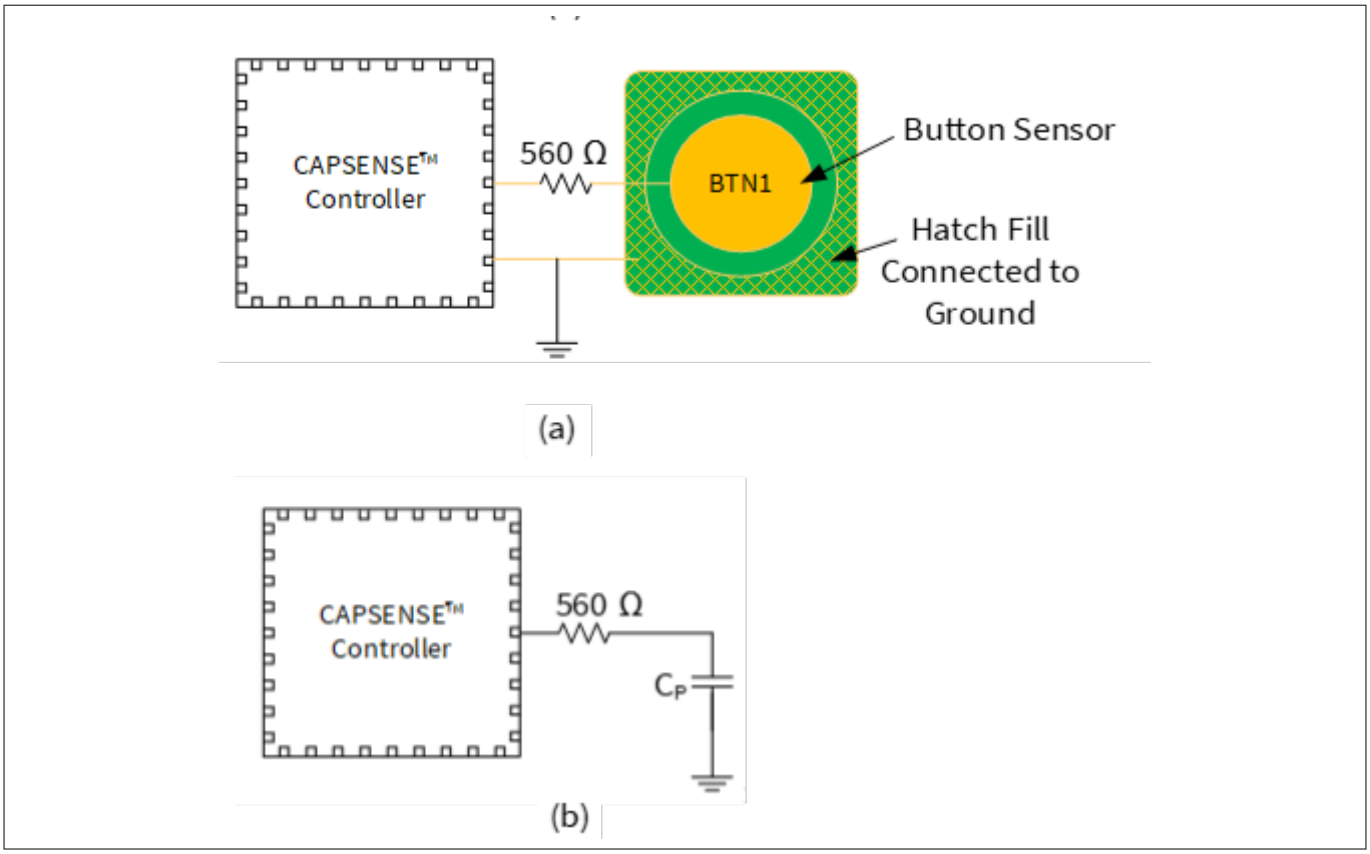


図 24　　標準的 CAPSENSE™システムのレイアウト

図 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) 、図 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。

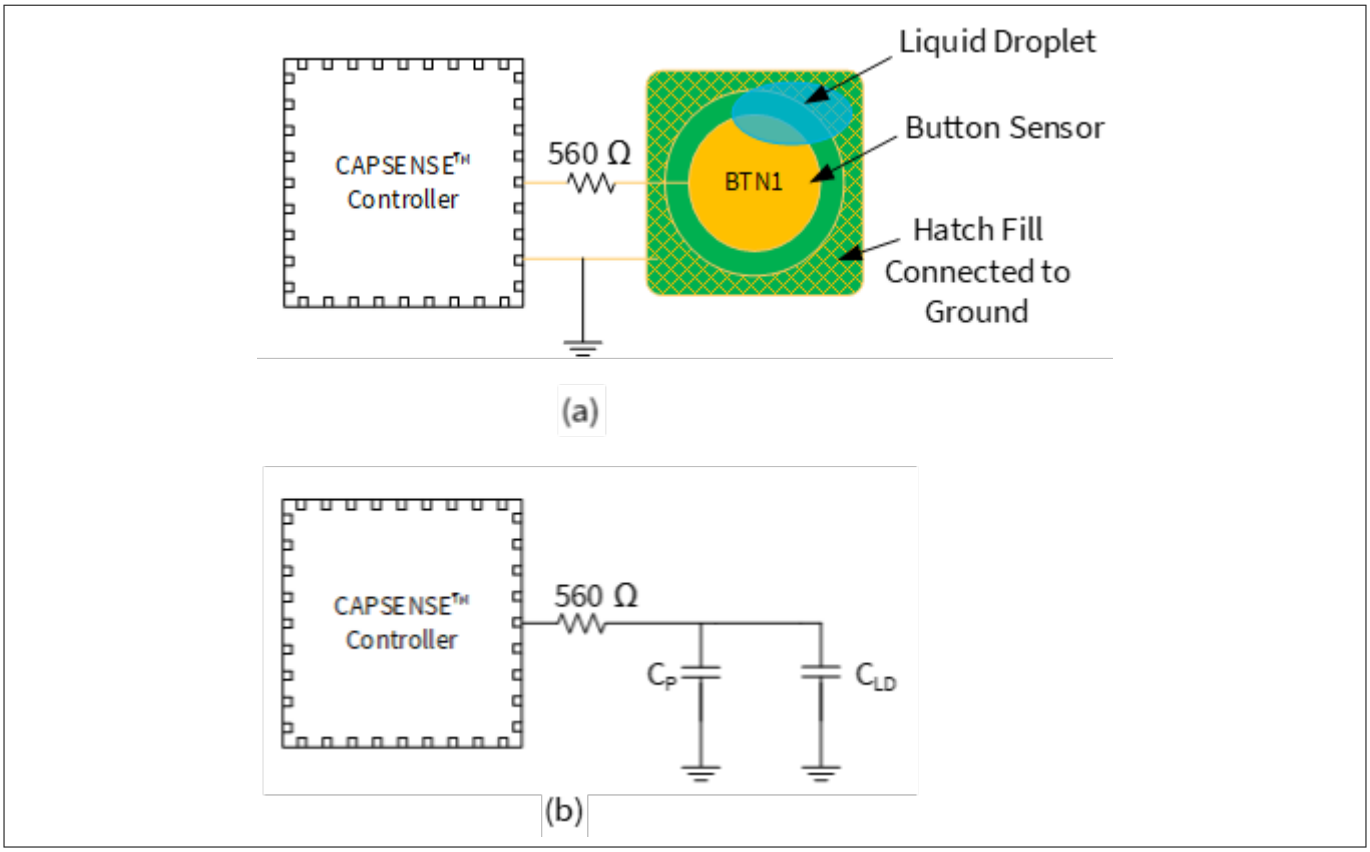


図 25　　ハッチが GND に接続されているときの液滴により追加された静電容量

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

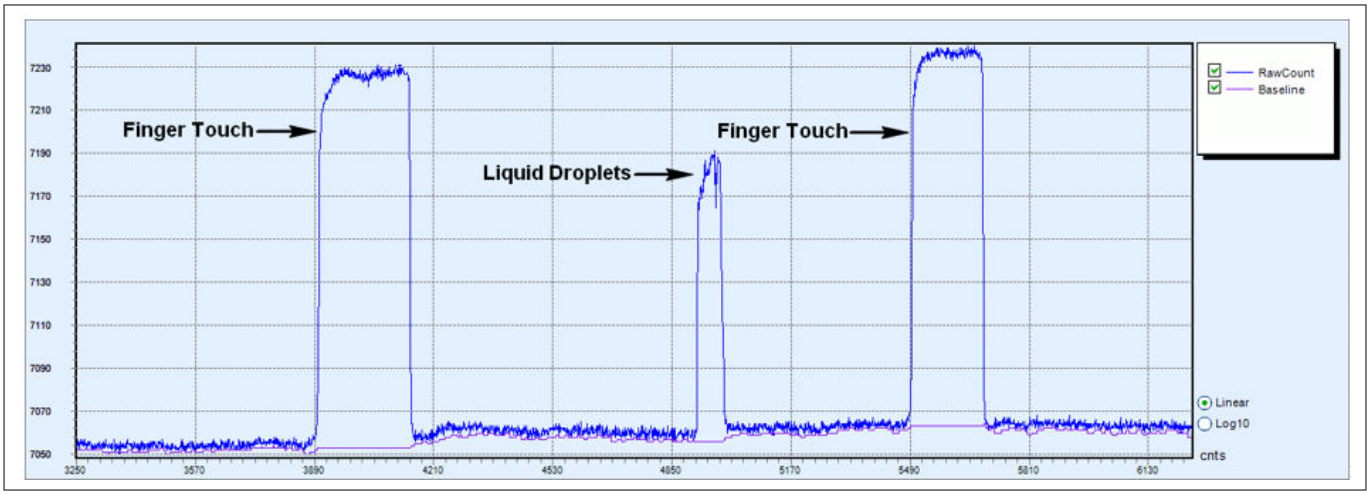


図 26　　センサー周囲のハッチが GND に接続されているときの液滴の影響

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。

図 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、図 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。

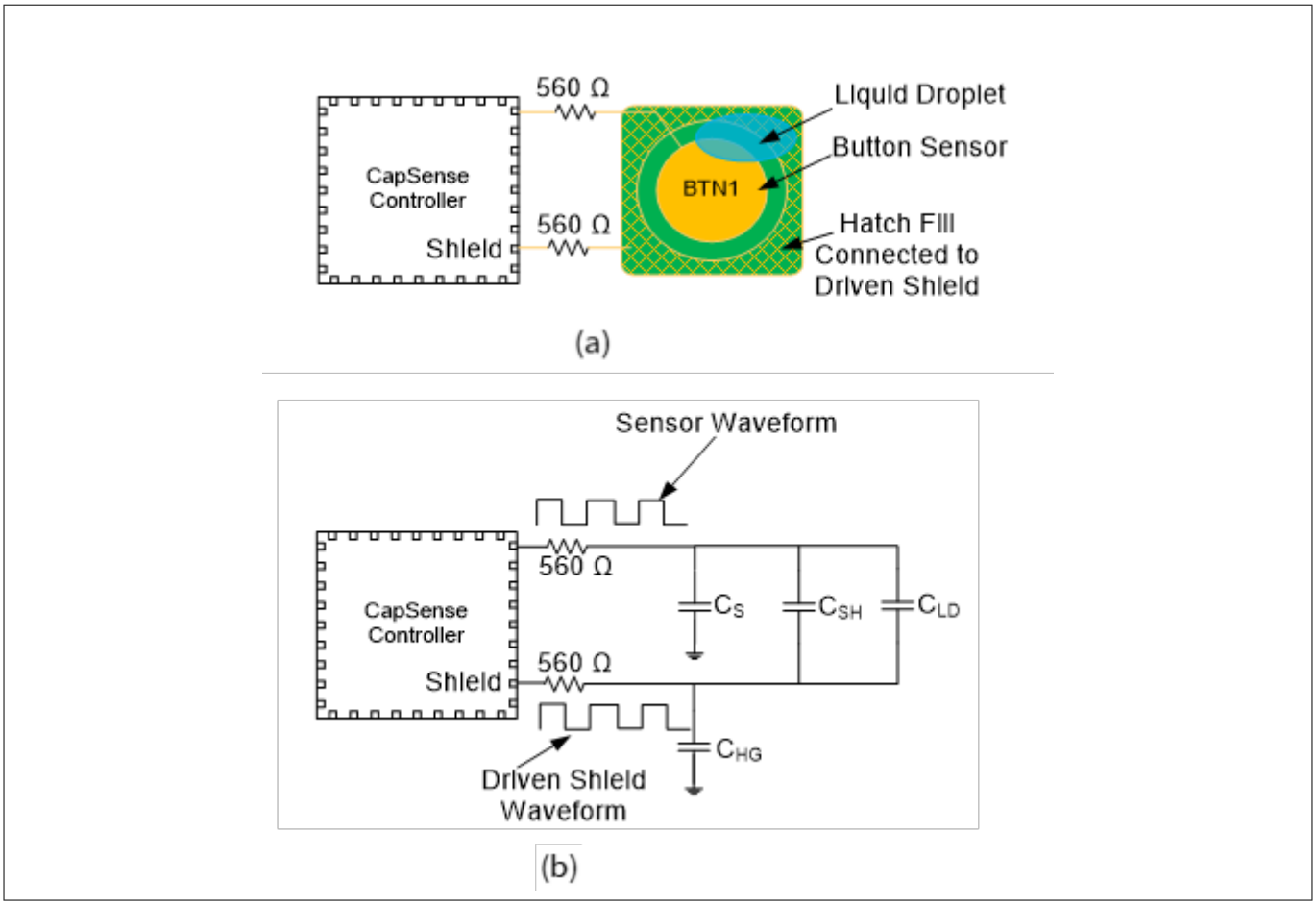


図 27　　センサー周囲のハッチがシールドに接続されているときの液滴により増加した静電容量

C_S = センサー寄生容量
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

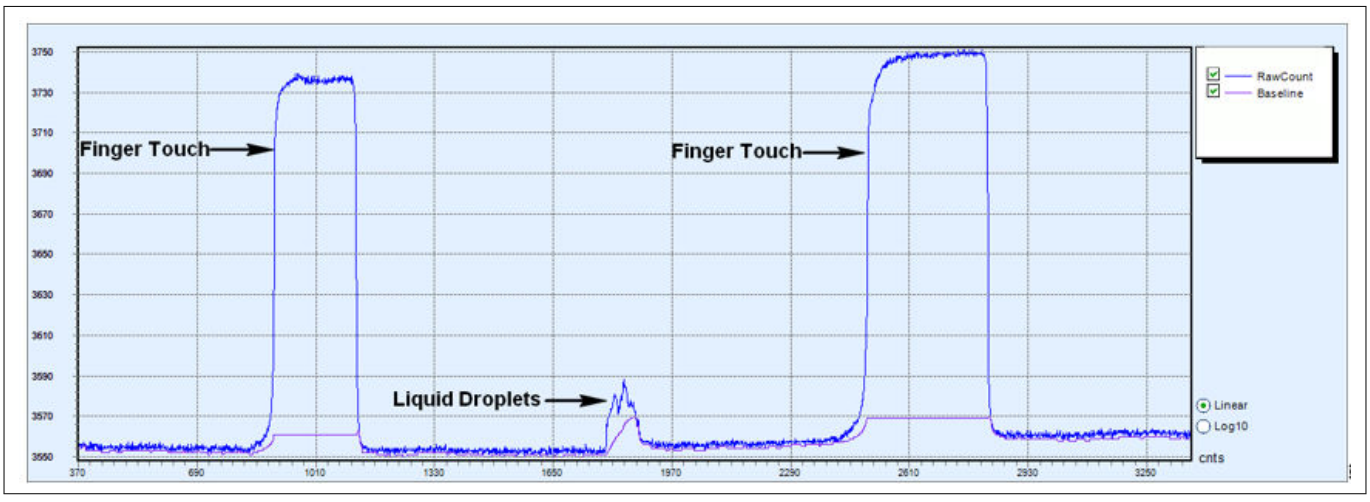


図 28　　センサー周囲のハッチが被駆動シールドに接続されているときの液滴の影響

図 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。図 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。

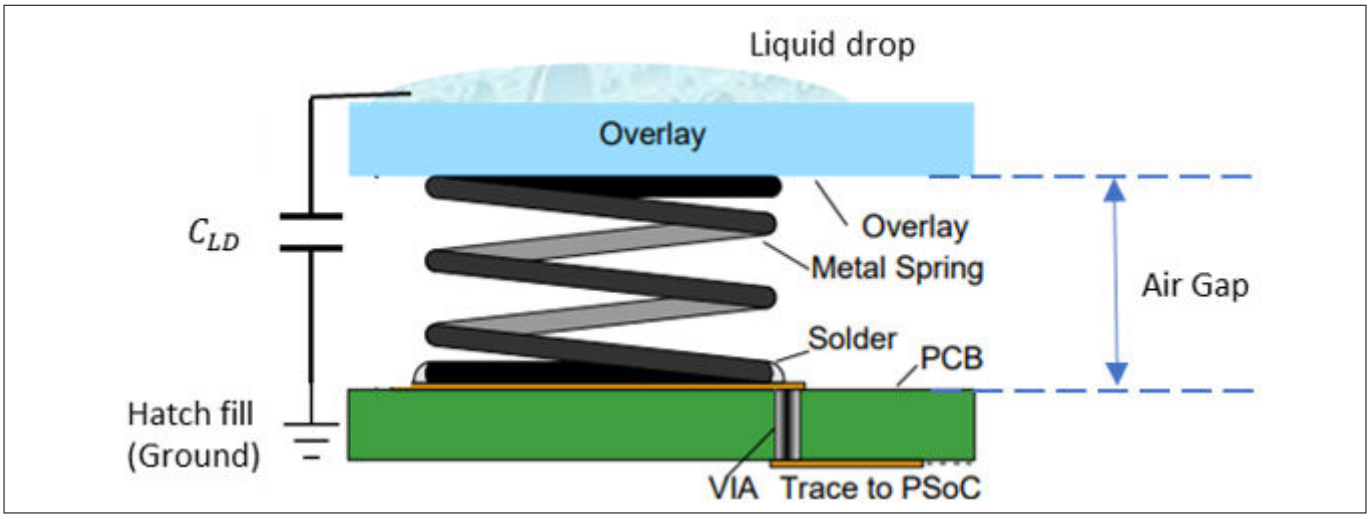


図 29　　スプリングセンサーの液滴によって追加された容量

2.5.1.2　被駆動シールド信号およびシールド電極

図 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。

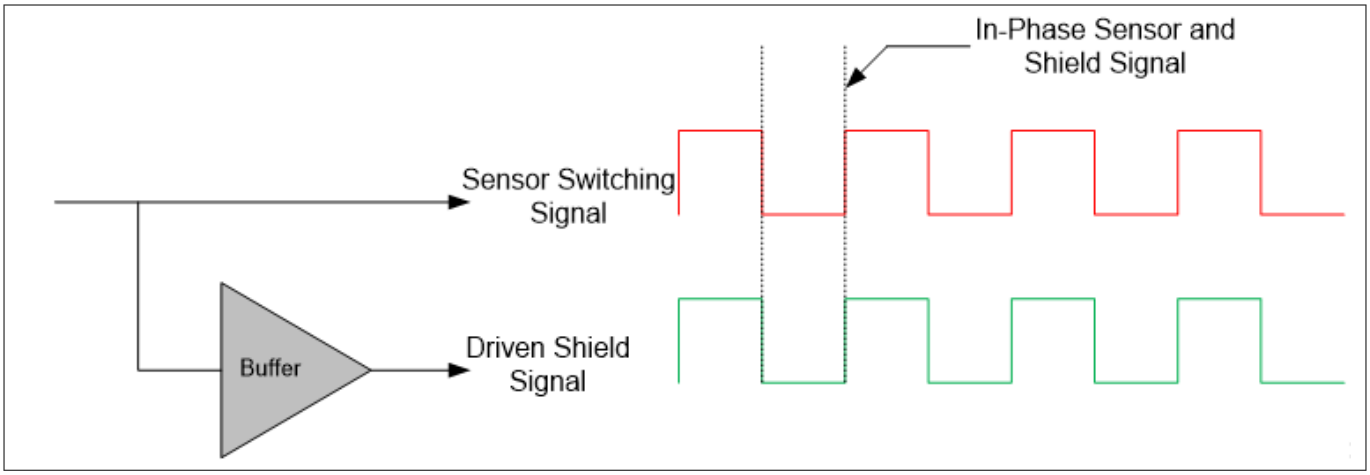


図 30　　被駆動シールド信号

- 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。CAPSENSE™入門の「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、図 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で、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。

注: 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。

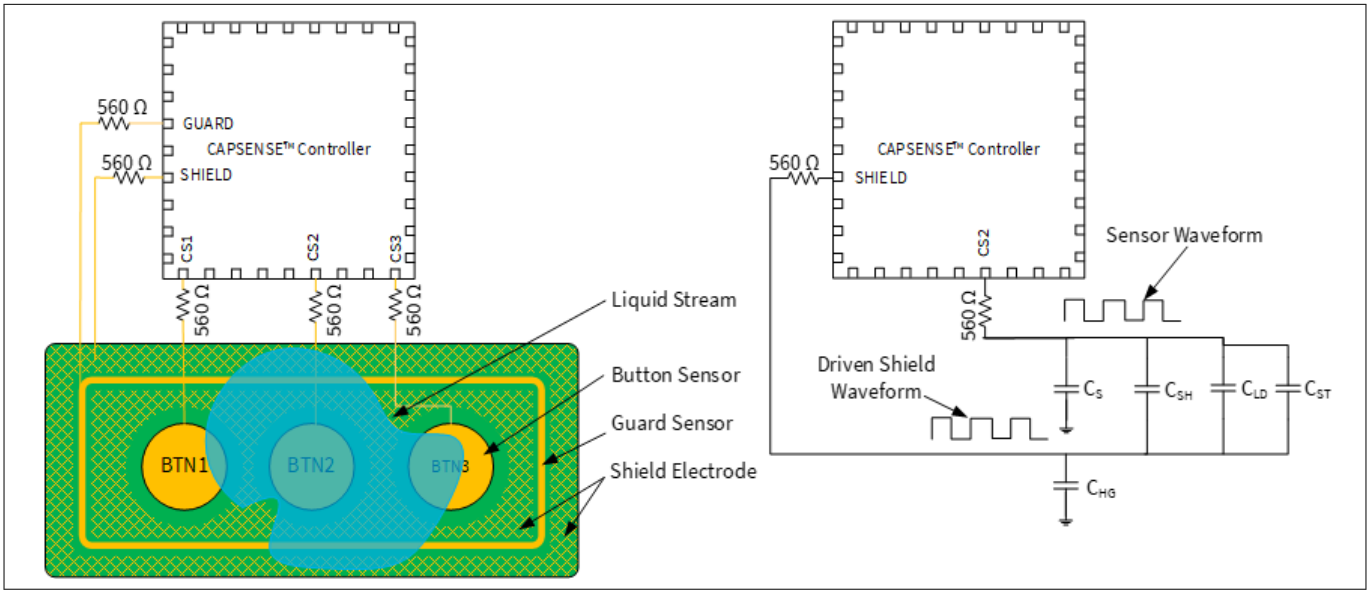


図 31 液体流がある場合の静電容量測定

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。

注: 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。

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、および図 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、図 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。

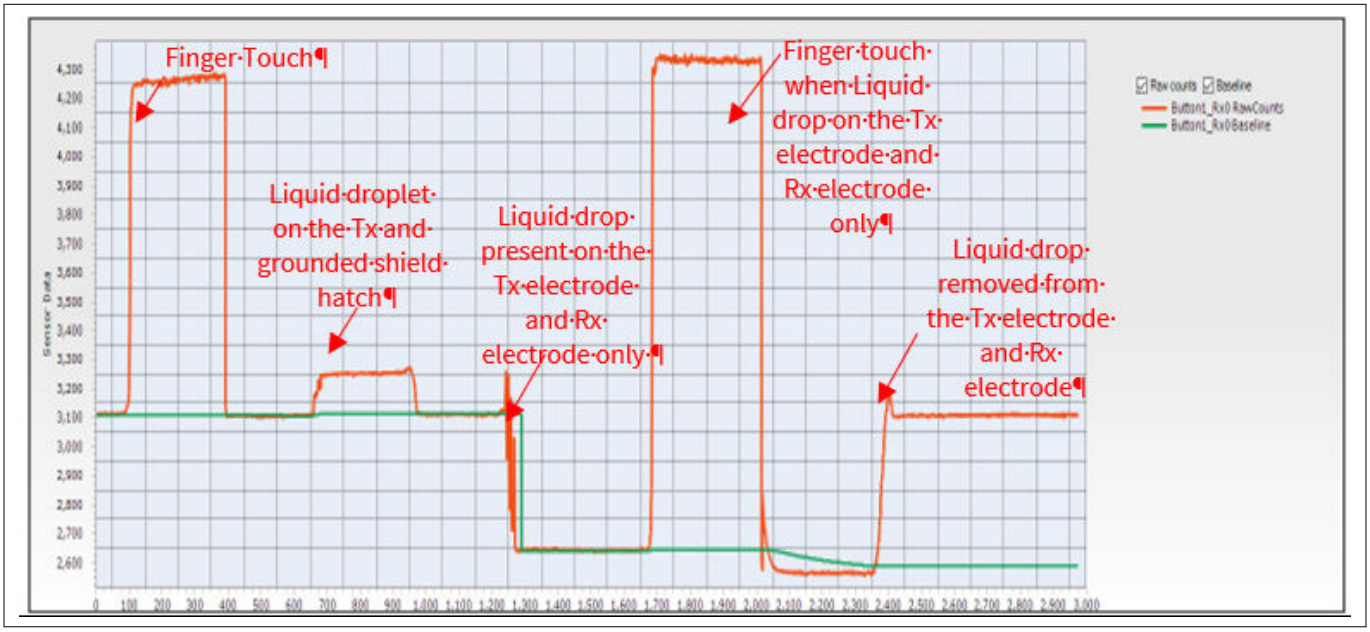


図 32 センサーの周りのハッチフィルがアースに接続される場合の CSX センサーに対する液滴の影響

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、低レベル API 関数 `CapSense_SetPinState()`を使用することによって、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、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図 33 に要約します。

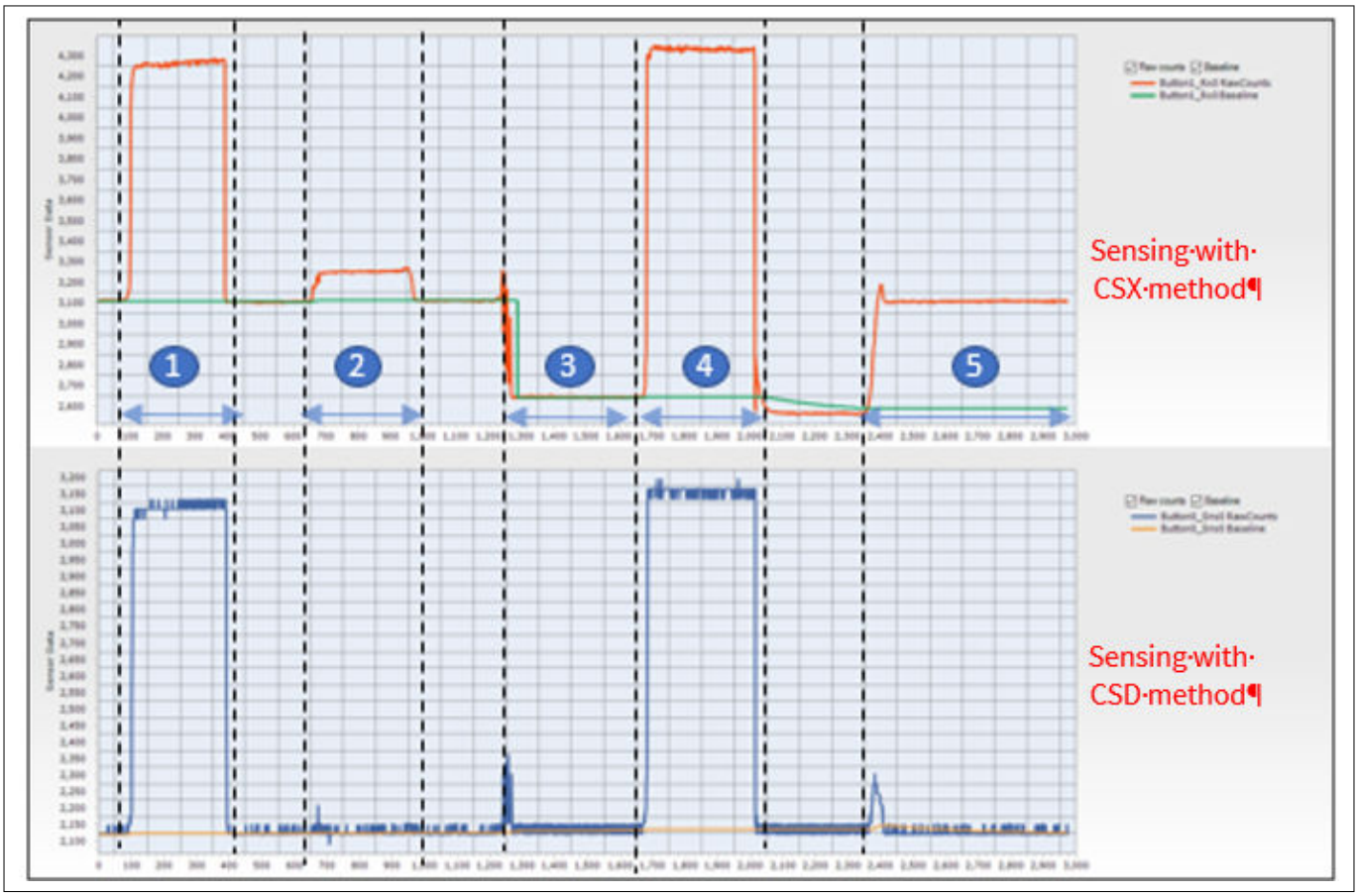


図 33　　CSD および CSX 方式でスキャンされた CSX センサーパターンに対する水滴の影響

図 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。図 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

図 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 ドキュメント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、石鹸水、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。石鹸水を用いて行ったテストの結果は石鹸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™システムは石鹸水の存在下でも動作できます。

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。図 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。

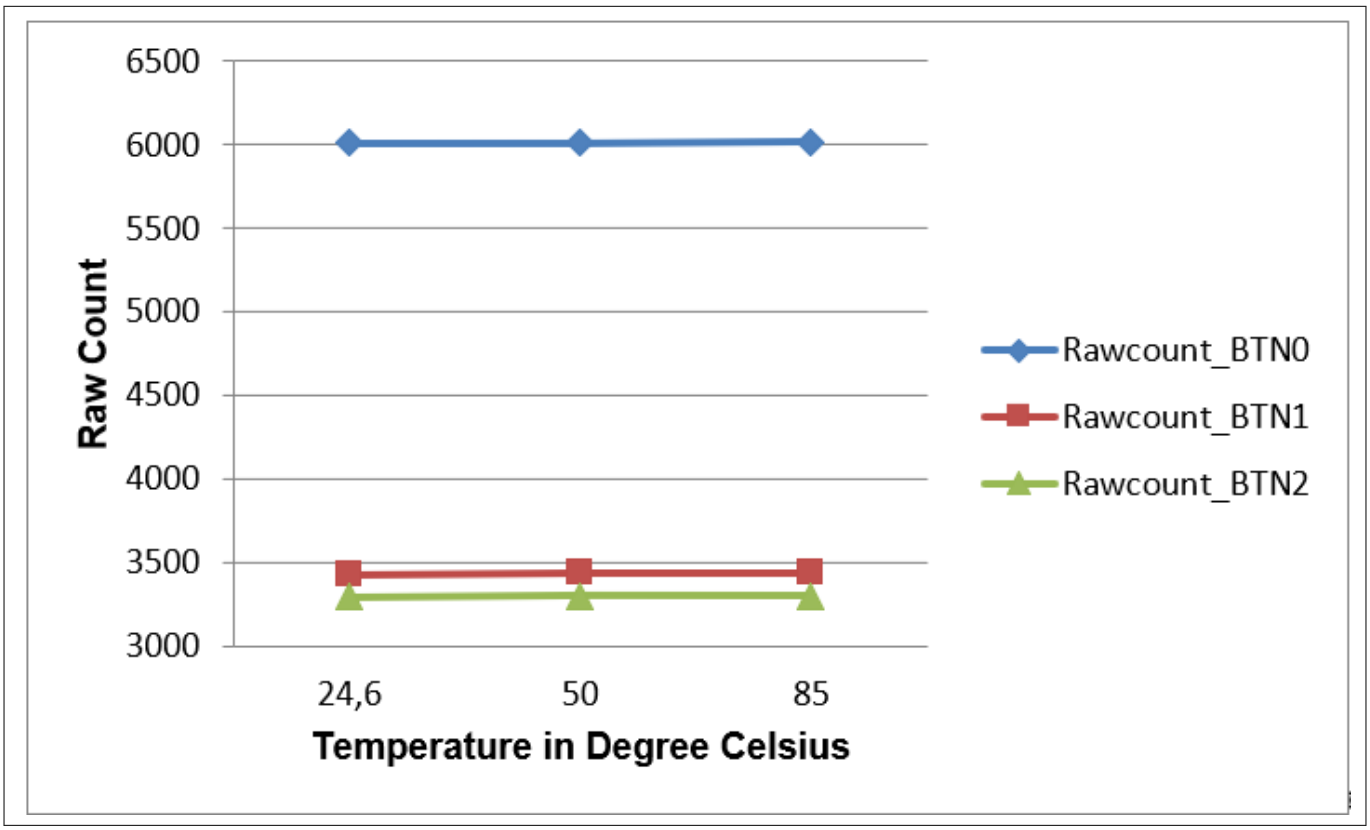


図 34 水の温度と raw カウントの変化

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 - PSOC™ 4 入門または 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。詳細については、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。表 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

表 2 に、CSD と CSX の CAPSENSE™アーキテクチャの主な違いを示します。

表 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} = 5 V; firmware filter なし; C_p ~= 33 pF; C_f = 0.1 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} = 5 V; firmware filter なし; C_p ~= 33 pF; SNR = 5:1;

(続く)

表 2 (続き) 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
ノイズフロア (rms)		–	–	500 aF for CSD-RM 100 aF for CSX-RM	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} - 5 V; C_p ~= 33 pF; C_M = 5 pF
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 ~= 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 [1] これは、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 (V_{ref})	1.2 V	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。	

(続く)

表 2 (続き) 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
	V_{DD} ノイズ影響	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。	
センスクロック周波数	自己容量	45 kHz ～ 6 MHz	45 kHz ～ 6 MHz	45 kHz ～ 6 MHz	センスクロック周波数が高いほど、低 C_p 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 kHz	45 kHz ～ 3 MHz	45 kHz ～ 6 MHz		
マルチチャネルサポート		No	No	Yes	'n'チャ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		--	--	1.2 nF	–	–
デバイスファミリ		PSOC™ 4100/4200 PSOC™ 4100 M/ 4200 M PSOC™ 4100 L/4200 L PSOC™ 4100 BL/ 4200 BL	PSOC™ 4000 PSOC™ 4000S PSOC™ 4100S PSOC™ 4100S Plus PSOC™ 6	PSOC™ 4100S Max PSOC™ 4000T(第 5 世代低消費電力)	–	–

1) 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。

3.2 Capacitive Sigma-Delta (CSD) センシング方式 (第 3 および第 4 世代)

図 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。

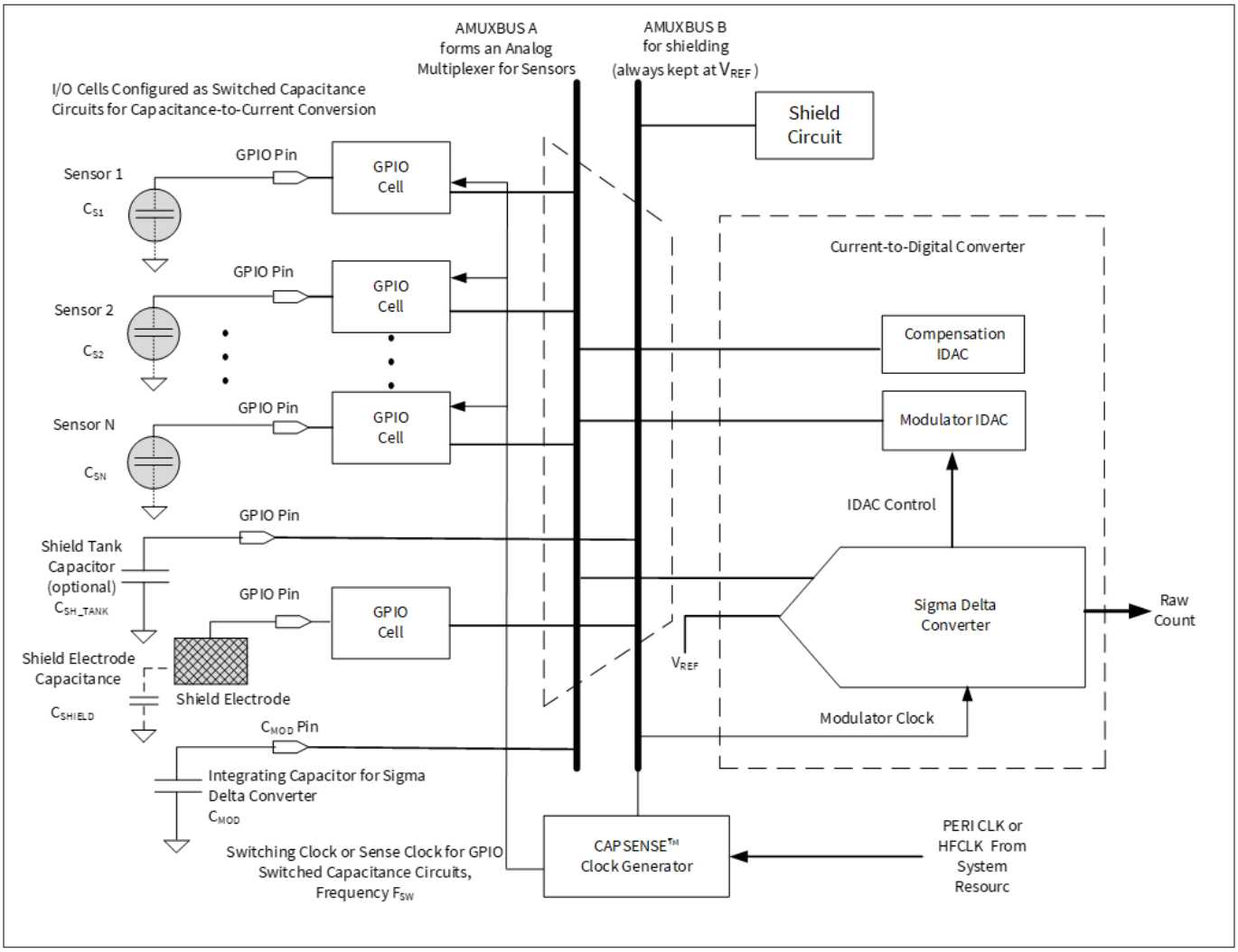


図 35 CAPSENSE™ CSD センシング

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ーであり、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。図 36 に GPIO セル構造の概略図を示します。

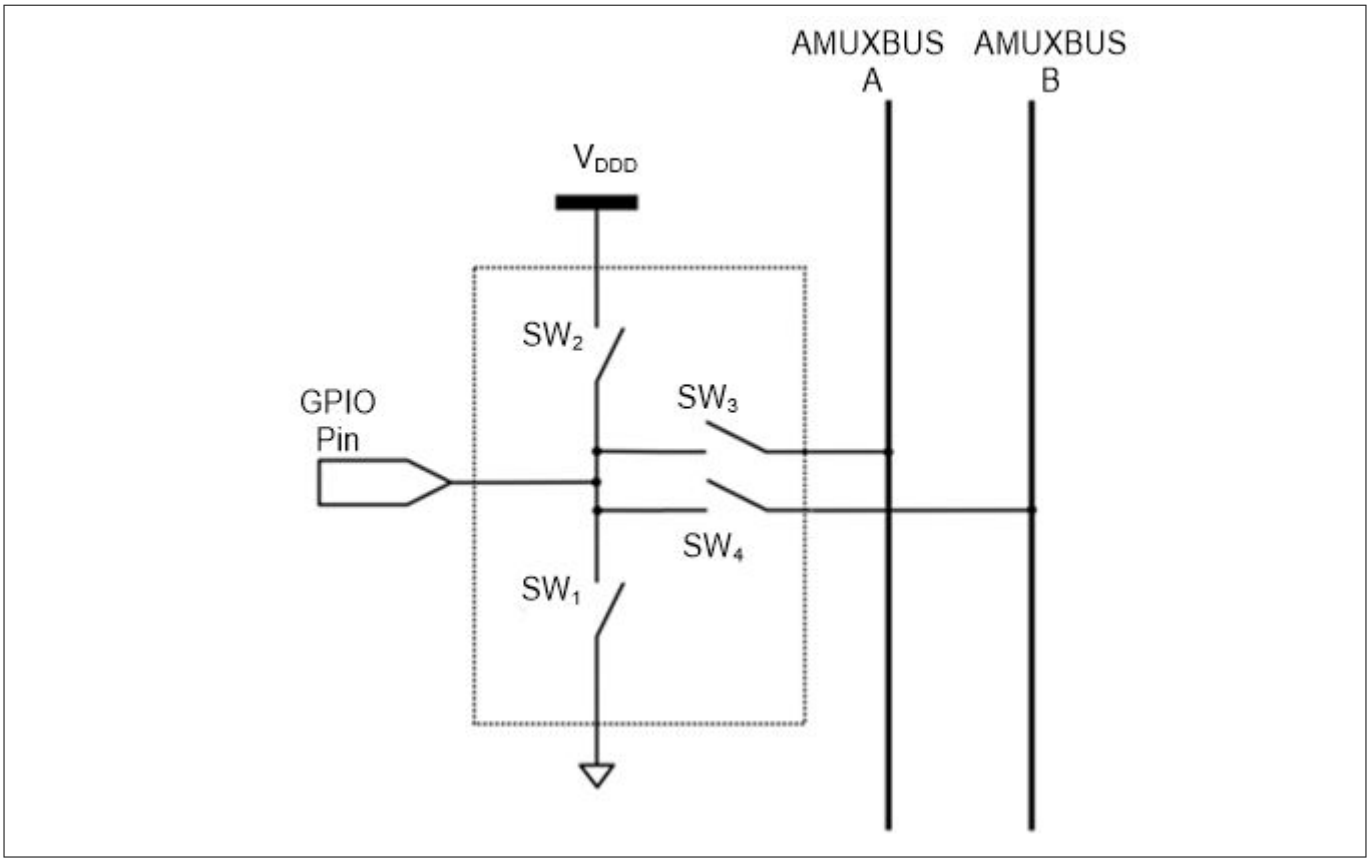


図 36 GPIO セル構造

PSOC™ 4 および PSOC™ 6 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

図 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。

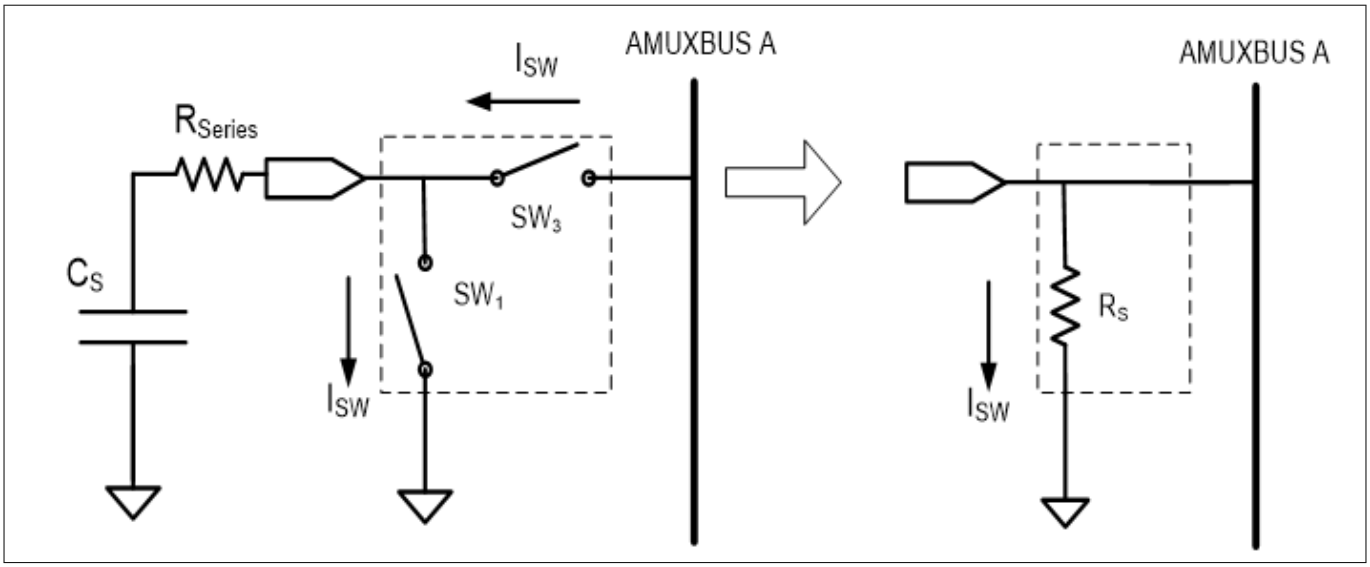


図 37　　GPIO セルの AMUXBUS A からの電流シンク

図 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 SSW_3 を制御します。図 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。

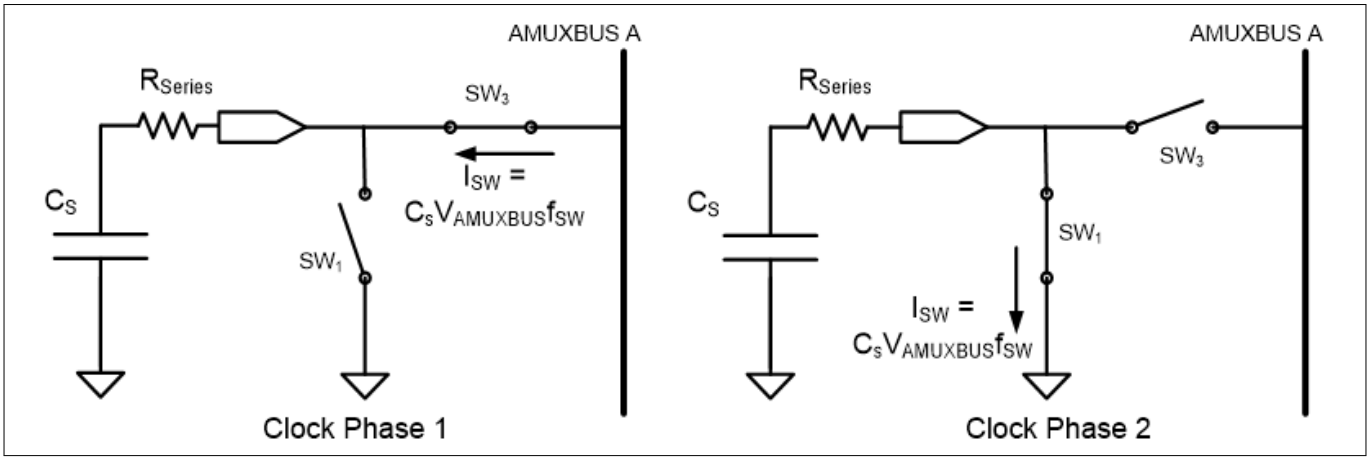


図 38　　重複しない SW_1 と SW_3 スイッチ

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、グランドに完全に放電するために時間 ($T_{SW}/2$) は十分です。図 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。

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

式 5　　センサーの等価抵抗

ここで、

C_S = センサーの静電容量

F_{SW} = センスクロック周波数

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)。図 39 は C_S 両端における電圧波形を示します。

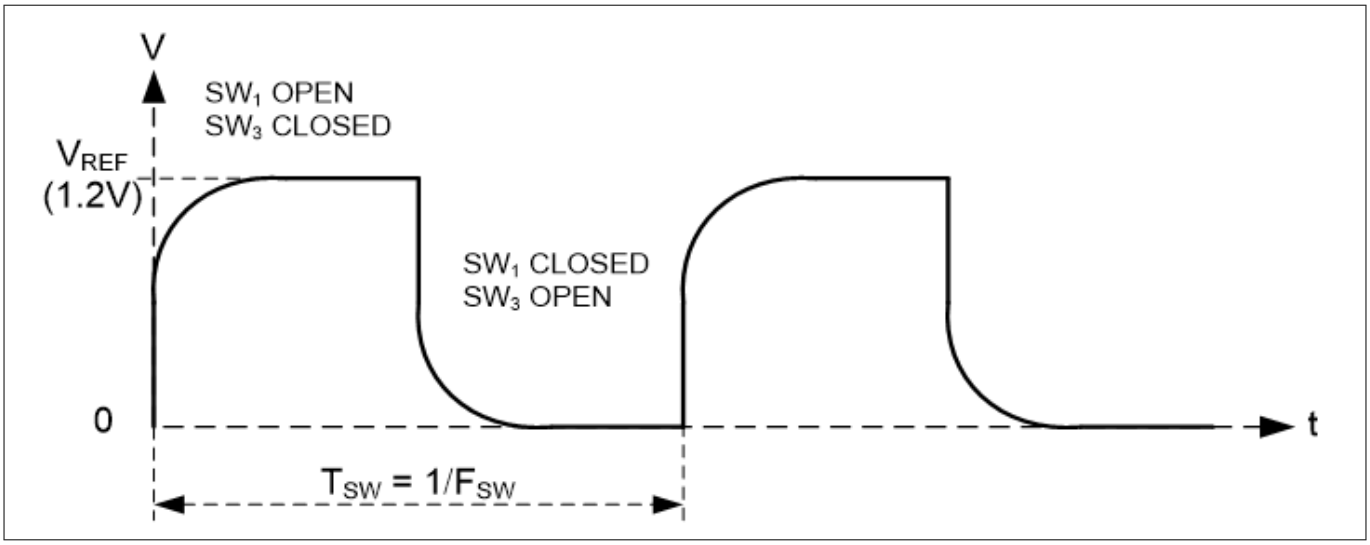


図 39　　センサー静電容量の両端における電圧

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。

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

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 (ICS)

3.2.3　IDAC シンク モード

IDAC シンク モードでは、図 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。図 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。

IDAC ソース モードに比べて、このモードが直接 V_{DDD} 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_{DDD} とともにこのモードを使用することを推奨します。

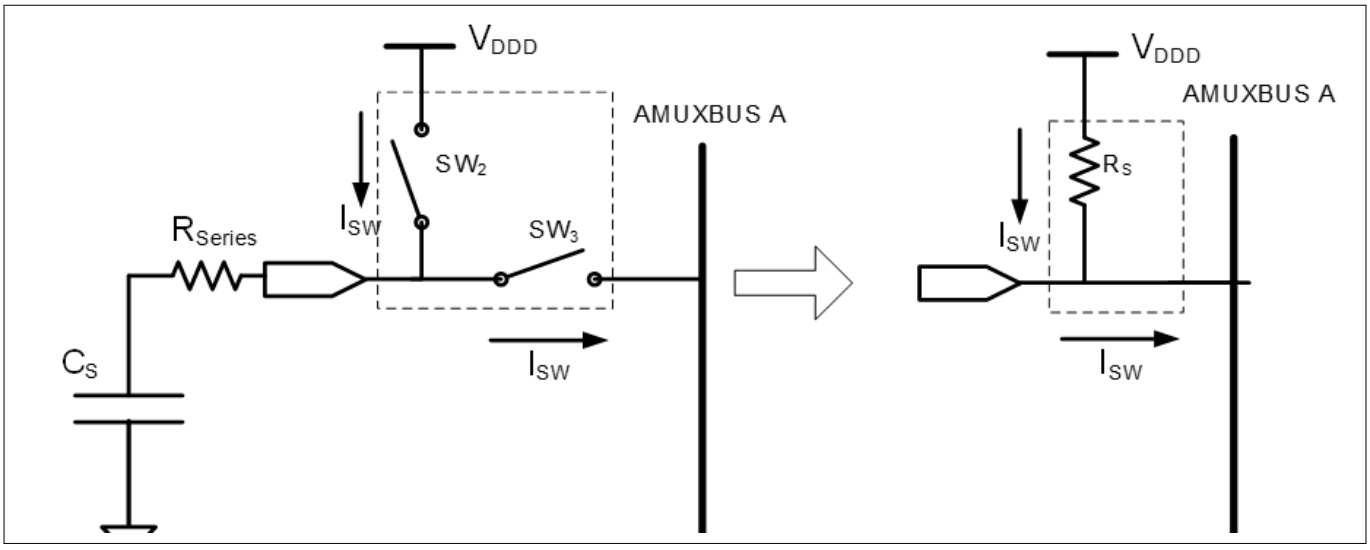


図 40　　GPIO セルの AMUXBUS A への電流ソース

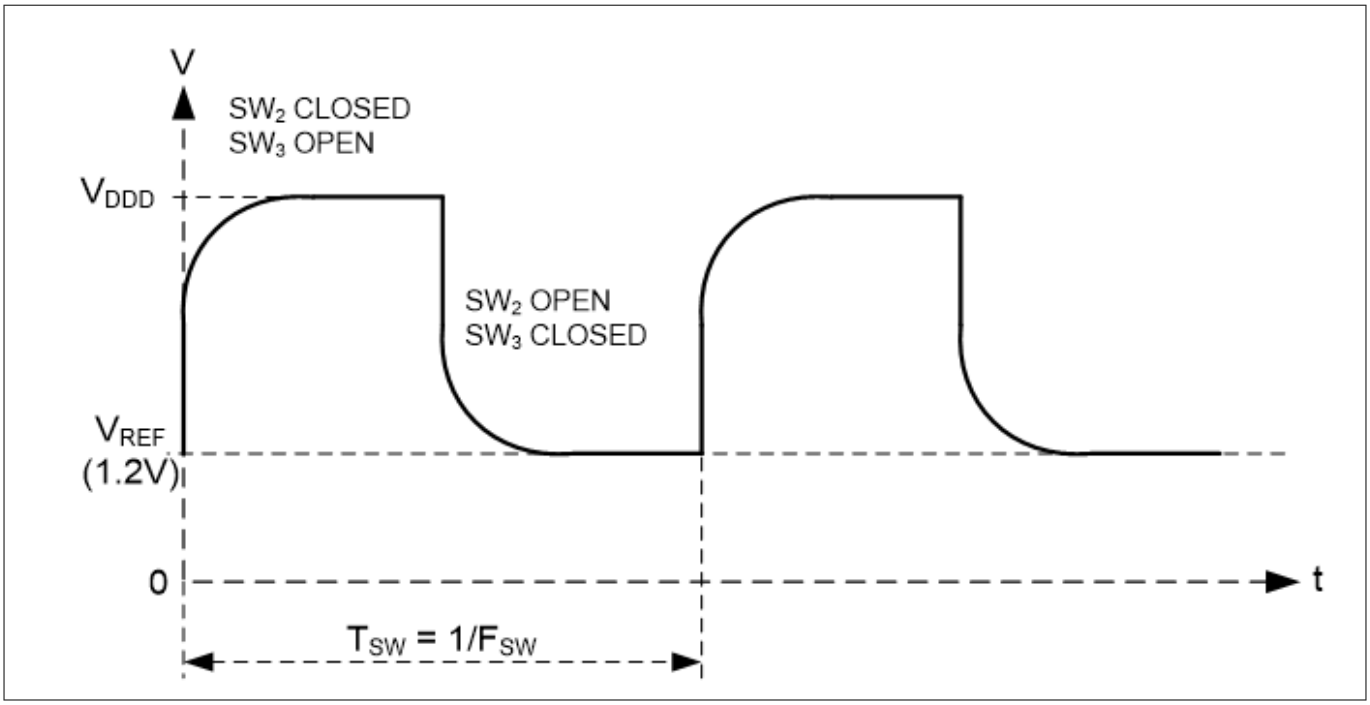


図 41　　センサー静電容量の両端における電圧

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。

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

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 (ICS)

3.2.4　CAPSENSE™クロック ジェネレーター

図 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} および変調器クロック 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 セルの容量から電荷へのコンバーターのために、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 (SSCx) モードもサポートします。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 ドキュメント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チューニングを参照してください。

3.2.4.2 変調器クロック

変調器クロックは、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。

$$\text{Sensor scan time} = \text{Hardware scan time} + \text{Sensor Initialization time}$$

式 8 センサー スキャン時間

$$\text{Hardware scan time} = \frac{\left(2^{Resolution} - 1\right)}{Modulator\ Clock\ Frequency}$$

式 9 ハードウェア スキャン時間

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、図 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、図 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} の推奨値を表 38 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 セルの容量から電荷へのコンバーターは、C_{MOD} 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 セルの容量から電荷へのコンバーターは、C_{MOD} 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。

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。

$$\text{raw count} = \left(2^N - 1\right)\frac{V_{REF}\,F_{SW}}{I_{MOD}}\,C_S$$

式 10 シングル IDAC ソースの raw カウント

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。

$$\text{raw count} = \left(2^N - 1\right) \frac{(V_{DD} - V_{REF})\, F_{SW}}{I_{MOD}}\ C_S$$

式 11　　シングル IDAC シンクの raw カウント

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、補償 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。

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

式 12　　デュアル IDAC ソースの raw カウント

IDAC シンク モードでの raw カウントは式 13 で計算されます。

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

式 13　　デュアル IDAC シンクの raw カウント

注: *raw カウントの値は常に正です。IDAC シンク モードに対して I_{COMP} が $(V_{DD}\text{-}V_{REF})C_S F_{SW}$ より小さく、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実に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 パラメーターと上記の式に示されたパラメーターの関係を表 3 に示します。

表 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 ドキュメント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 ドキュメント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 ソース	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。
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。

(続く)

表 3 (続き) 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
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	変調器クロック周波数	変調器クロック分周器は raw カウントの式に影響しません。 詳細については変調器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。図 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 (図 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については、被駆動シールド信号およびシールド電極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_{DDD} またはグランド) 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。

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

図 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、表 38 を参照してください)。

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

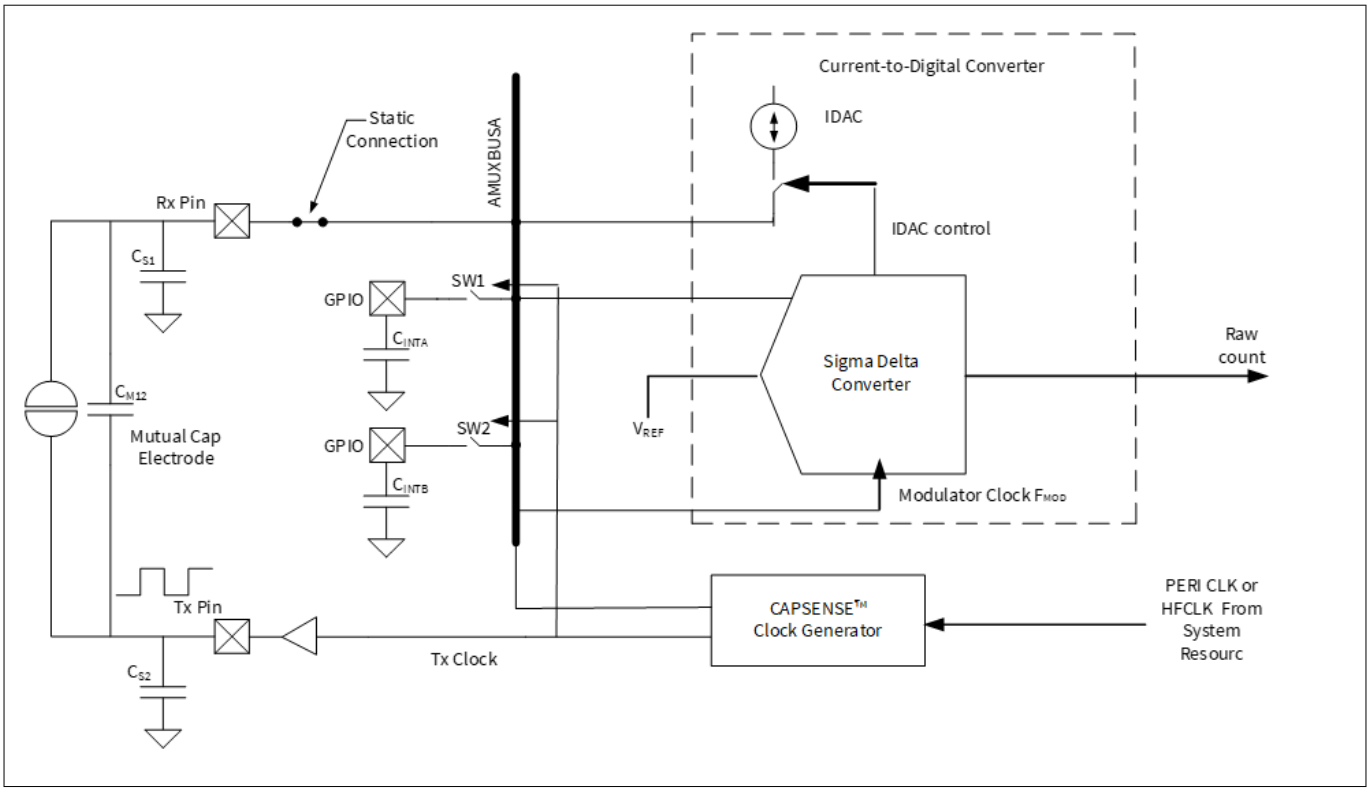


図 42　　CAPSENSE™ CSX センシング方式の構成

CSX センシング方式では、図 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、V_{DDIO} (または指定された部品番号で V_{DDIO} が使用できない場合は V_{DDD}) とグラ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表 38 に示します。

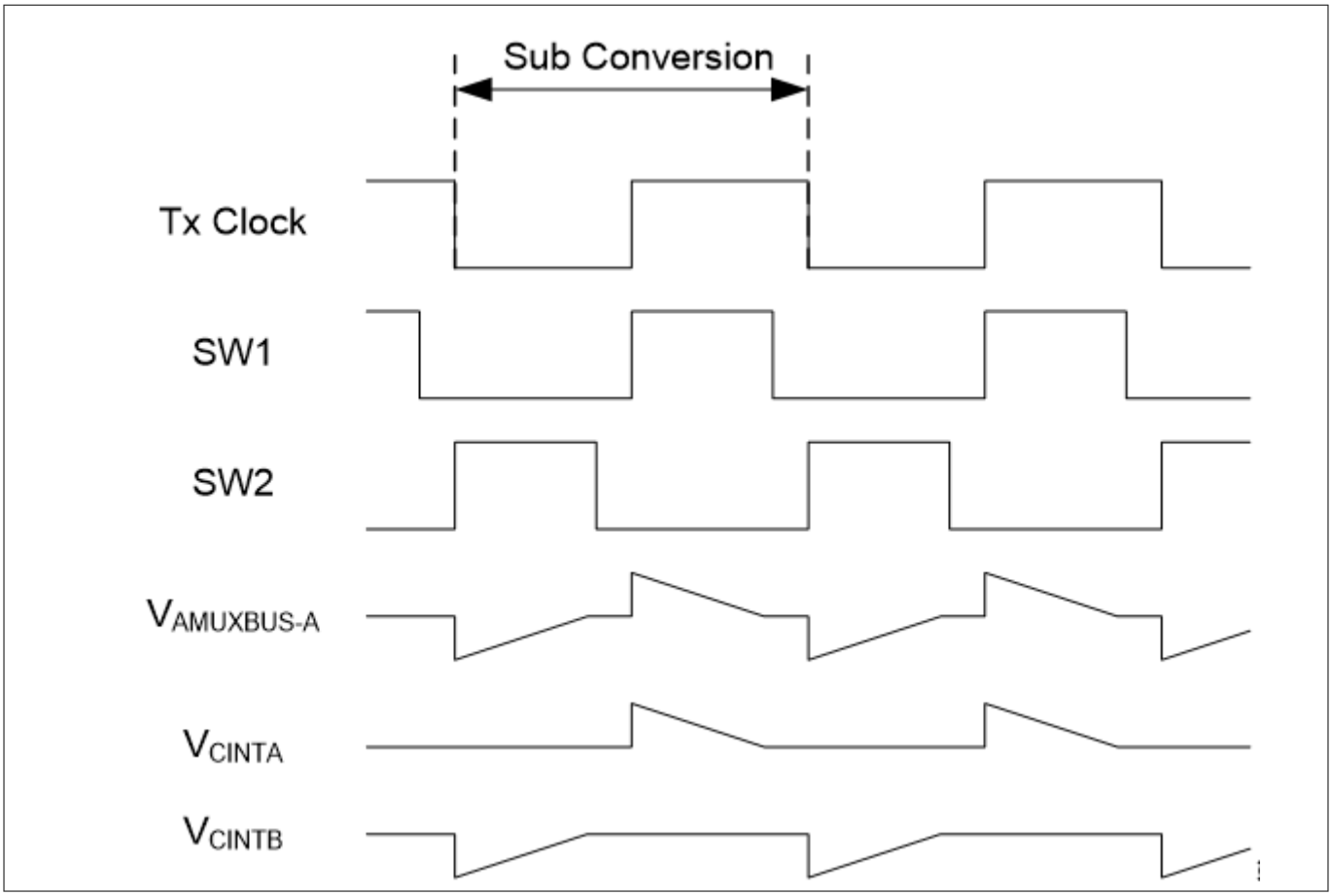


図 43　　CSX センシング波形

図 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。Tx クロックの立ち下りエッジで、SW2 が閉じ (このときに SW1 が開いている)、充電電流が Rx 電極から Tx 電極に流れます。これにより、C_{INTB} の電圧は V_{REF} を下回ります。IDAC は C_{INTB} の電圧を V_{REF} に戻すためにソース 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。

変調器クロックは 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、変調器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。変調器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。

$$Rawcount_{Counter} = \frac{2\,V_{TX}\;F_{TX}\,C_M\,\mathrm{MaxCount}}{IDAC}$$

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

式 14　　相互容量センシングでの raw カウントの関係

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

F_{TX} = Tx クロック周波数

F_{Mod} = 変調器クロック周波数

N_{Sub} = サブ変換回数

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。

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

式 15　　$Rawcount_{Component}$ を決定する式

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 世代および第 5 世代低電力)

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。

3 PSOC™ 4 および PSOC™ 6 MCU CAPSENSE™

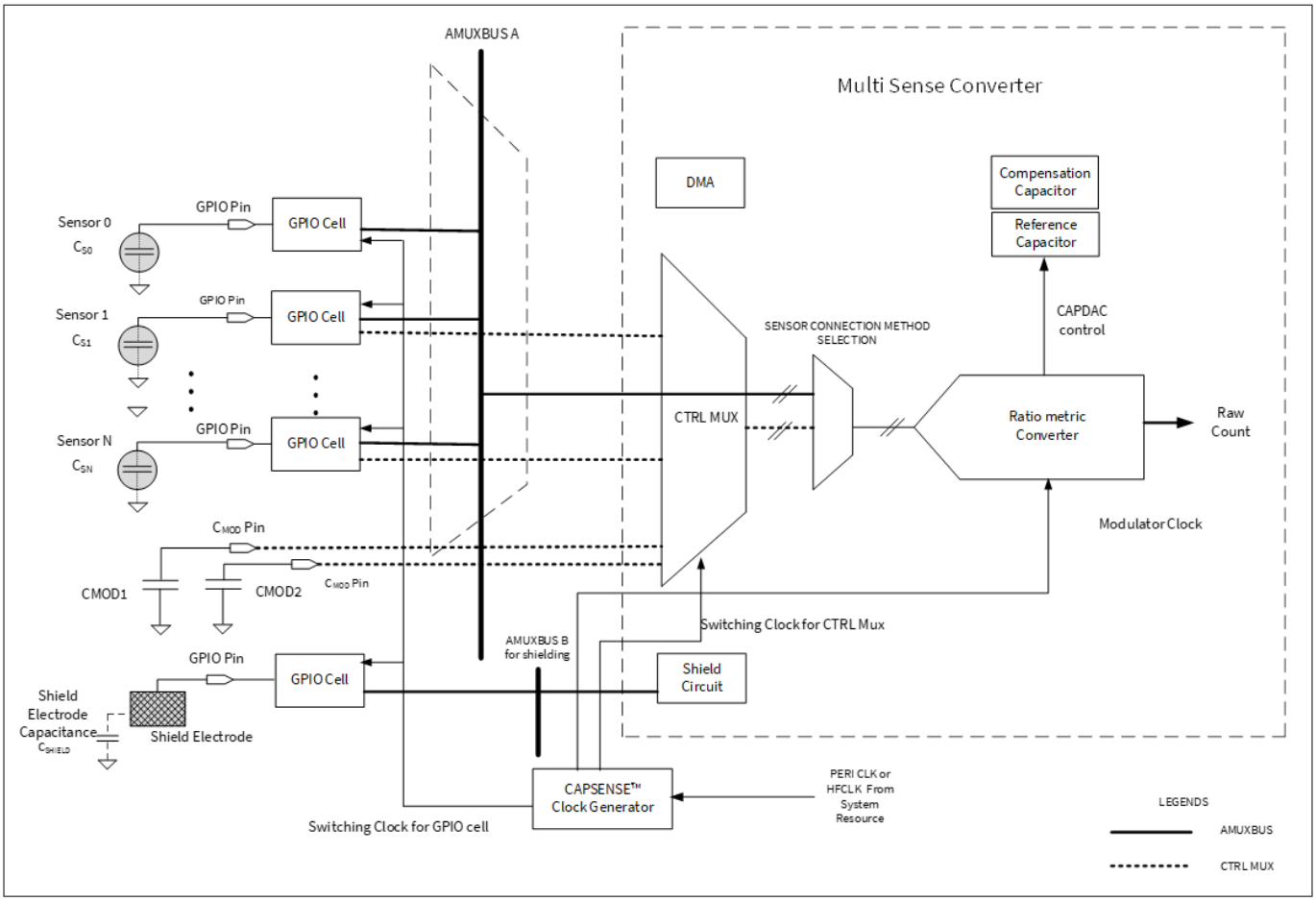


図 44　　CAPSENSE™ CSD-RM (第 5 世代)

第 5 世代の低消費電力 (マルチセンス ローパワーまたは MSCLP として知られる) 技術は、システム リソースへの依存を取り除いた内部クロックを備えています。自律スキャンは、CTRLMUX および DMA がなくても有効です。すべての AMUXBUS CAPSENSE™ ピンは自律スキャンをサポートします。

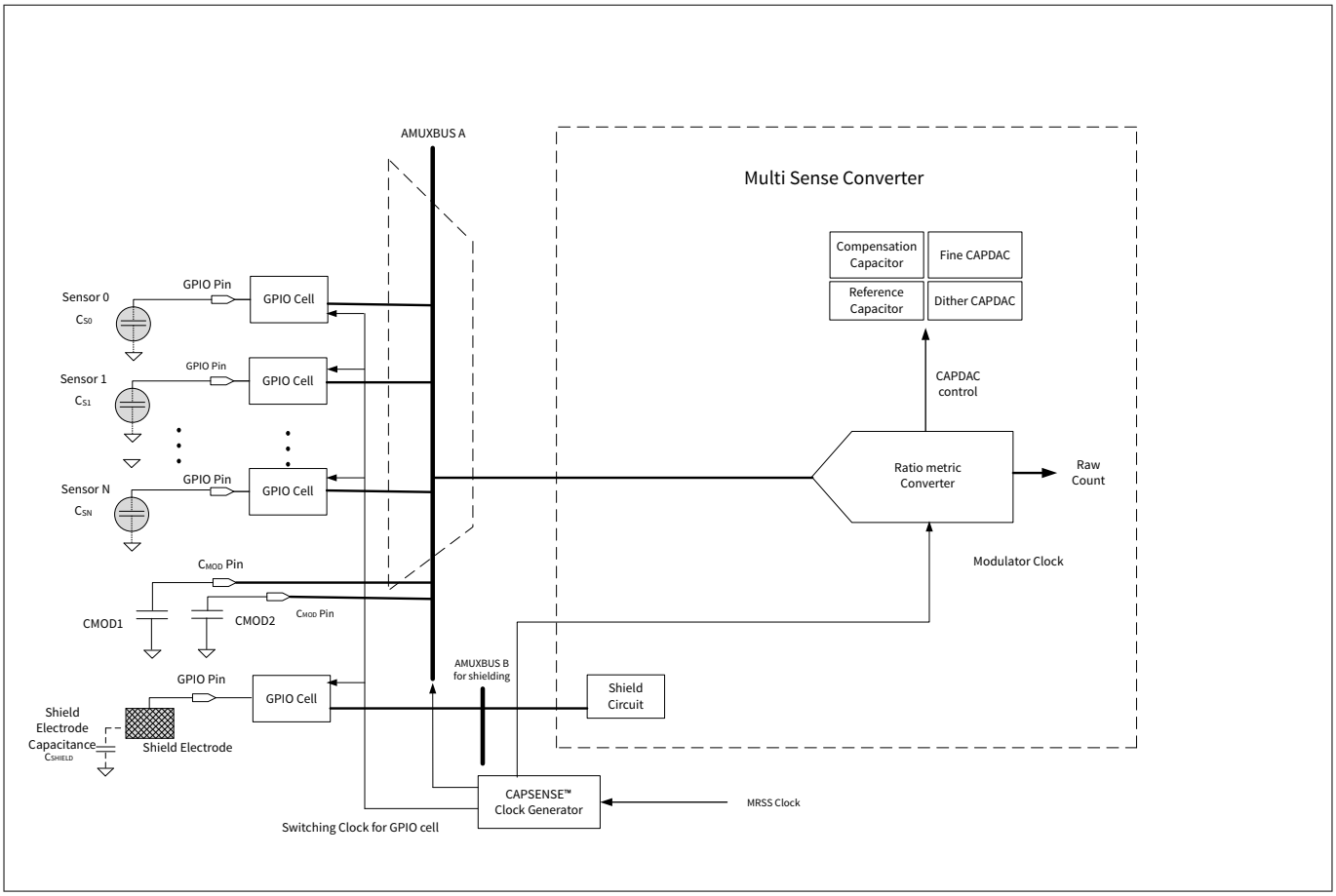


図 45　　CAPSENSE™ CSD-RM (第 5 世代低消費電力)

3.4.1　　GPIO セルの容量から電荷へのコンバーター

第 3.2.1 章では、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。図 46 に、GPIO セルの構造を示します。

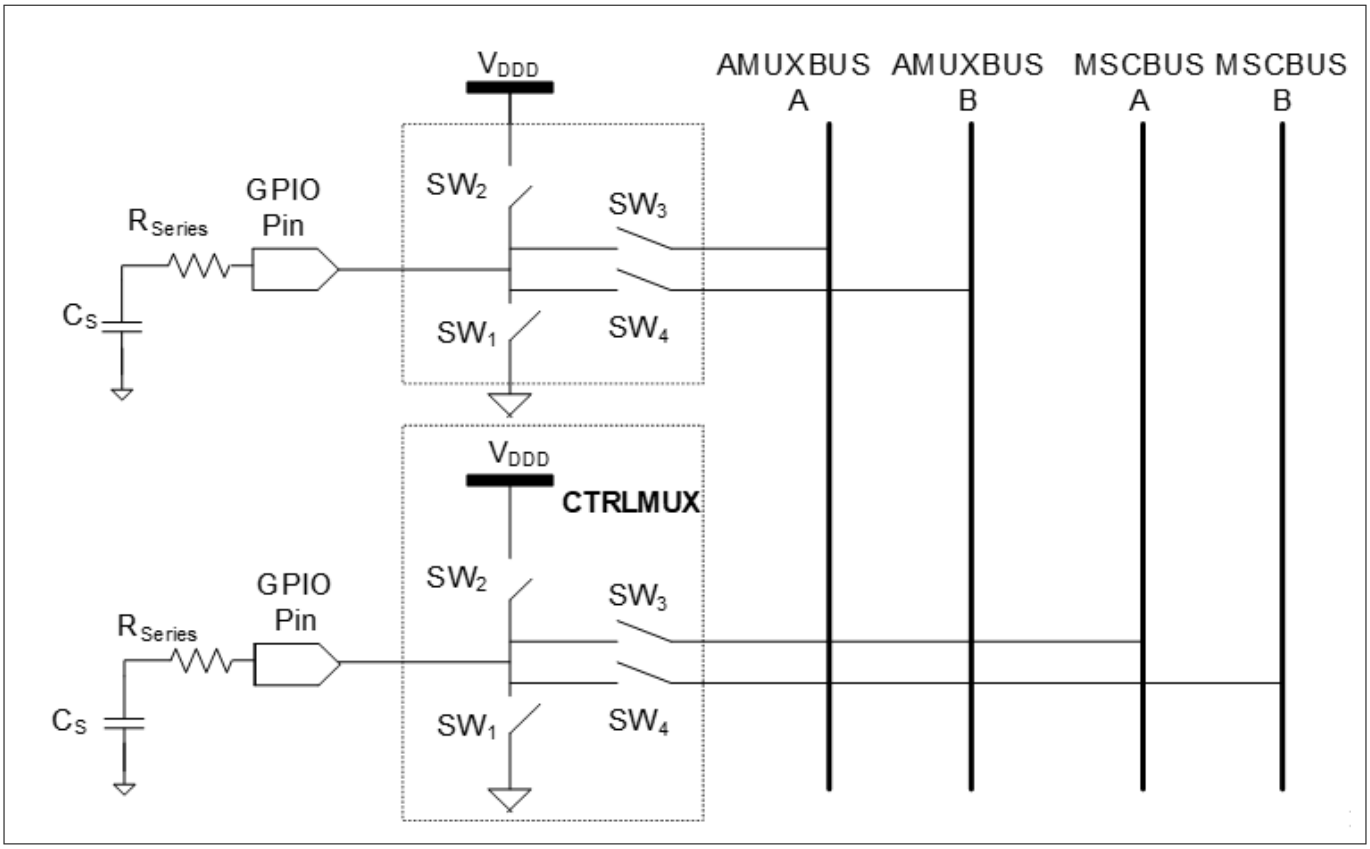


図 46　　GPIO セル構造 (第 5 世代)

第 5 世代の低消費電力アーキテクチャでは、センサーは AMUX にのみ接続され、AMUX は自律スキャンをサポートします (CTRLMUX よりも多くのピンをサポートします)。

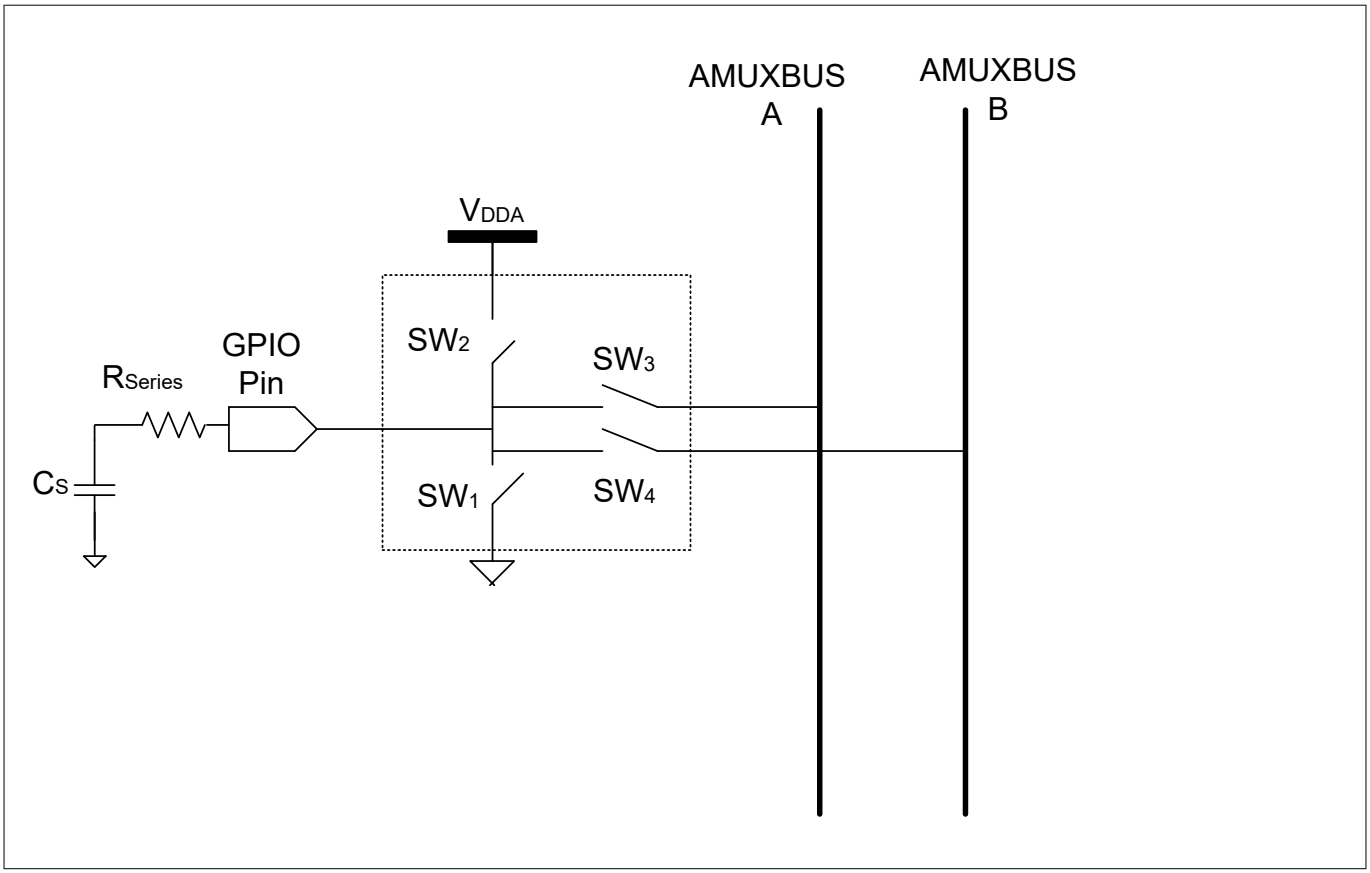


図 47　　GPIO セル構造 (第 5 世代低消費電力)

図 48 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。

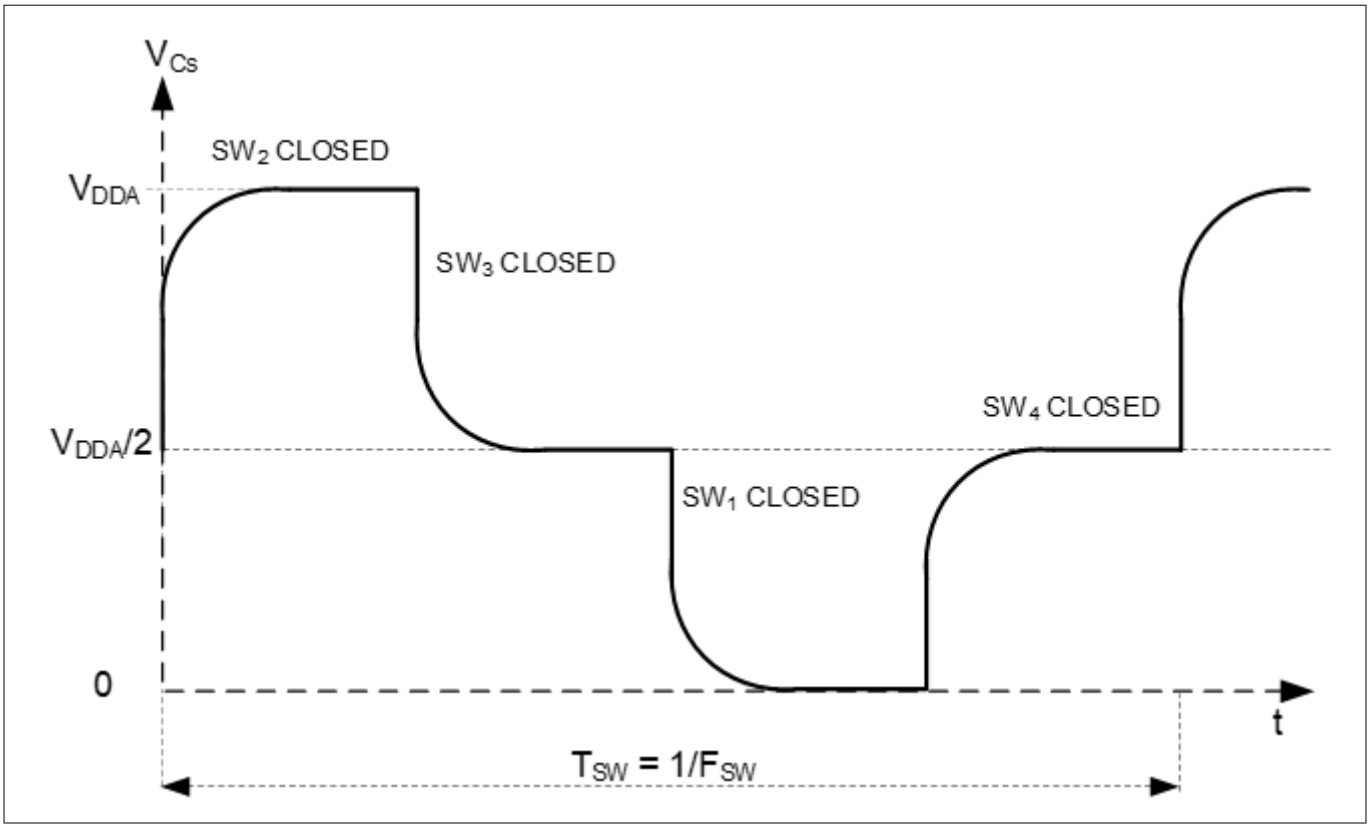


図 48　　センサー静電容量の両端における電圧

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。

第 5 世代の低消費電力 CAPSENSE™は、Fine CDAC と Dither CDAC で構成されます。Fine CDACFine 基準 CDAC (Cfine)はプログラマブル CDAC で、Reference CDAC の分解能をより細かくするために使用されます。CDAC ディザは、フラット スポット (またはデッドゾーン) を減らすために使用されます。

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} と変調器クロック 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、第 3.2.4.1 章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　変調器クロック

変調器クロックは、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 および式 17 に基づいてセンサーのスキャン時間を決定します。

$$\text{Sensor scan time} = \text{Hardware scan time} + \text{Sensor initialization time}$$

式 16　　センサー スキャン時間

$$\text{Hardware scan time} = \frac{(Number\ of\ subconversions)}{Sense\ clock\ frequency}$$

式 17　　ハードウェア スキャン時間

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、GPIO セルの容量から電荷へのコンバーターセンサー容量は、で説明されているように、C_{MOD}s 間の電圧のバランスを崩す方法で両方の C_{MOD}s 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}s 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}s 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。

- 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。

$$\text{Rawcount} = \text{Maxcount} \cdot \frac{C_S}{SnsClk_{Div} \cdot C_{ref}}$$

式 18　　CSD-RM シングル CDAC raw カウント

デュアル CDAC モードでは、補償 CDAC は常にオンです。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。

$$\text{Rawcount} = \text{Maxcount} \cdot \frac{C_S - 2 \cdot \frac{SnsClk_{Div}}{CompClk_{Div}} \cdot C_{comp}}{SnsClk_{Div} \cdot C_{ref}}$$

式 19　　CSD-RM デュアル CDAC raw カウント

ここで、

MaxCount = $N_{Sub}.SnsClk_{Div}$

N_{Sub} = サブ変換回数
$SnsClk_{Div}$ = 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
CIC2 を有効にすると、

$$\text{MaxCount} = \text{Decimation Rate}^2 * \quad K$$

式 20　　最大 raw カウント

ここで、

$$\text{Decimation rate} = \frac{\text{SnsClk}_{Div} * \text{N}_{sub}}{3}$$

式 21　　デシメーションレート

$$\text{K} = \frac{\text{No. of CIC2 samples}}{\text{CIC2 Hardware divider}}$$

式 22

可能な限り高い raw カウントを得るために、"K"を "1"に設定できるように、CIC2 のサンプル数を選択することを推奨します。

表 4　　CIC2 ハードウェア分割ルックアップテーブル

CIC2 サンプル数	分周器用推奨 CIC2 シフト	CIC2 ハードウェア分周器
2	1	2
3〜4	2	4
5〜8	3	8
9〜16	4	16
17〜32	5	32
33〜64	6	64
65〜128	7	128
129〜256	8	256

CAPSENSE™コンフィギュレーターで CIC2 シフトの自動オプションを選択すると、表 4 に従ってハードウェアシフト（ハードウェアデバイダ）が選択されます。

デシメーションレートはダウンサンプリングレートであり、最大 255 に達した時、CIC2 アキュムレータ(25 ビット)のオーバーフローを回避するために以下の条件を確保する必要があります。

$$\text{CIC2_Acc_Size}_{Max} \geq \text{TRUNC}\left(\frac{\text{N}_{sub} * \text{Sns_Clk_Div}}{\text{Decimation}} \text{-1}\right) * \text{Decimation}^2$$

式 23　　CIC2 アキュムレータのオーバーフロー状態

ここで、

$CIC2_Acc_Size_{Max}$ - CIC2 アキュムレータの最大サイズ

最大 CIC2 アキュムレータ サイズは、ダイレクト クロックの場合は $(2^{25} - 1)$、PRS クロックの場合は $(2^{25} - 1)/2$ です。

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。差動アーキテクチャのため、第 5 世代および第 5 世代の低消費電力 CAPSENSE™では、特定の電源リップルに対する raw カウントのドリフトが大幅に減少します。電源リップルがデータシートに記載されている範囲内であれば、第 5 世代および第 5 世代の低消費電力 CAPSENSE™技術を使用した設計に LDO は不要である可能性があります。

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、図 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 [4]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。

第 5 世代の低消費電力では、画像図 45 に示すように、CTRLMUX が削除されます。AMUXBUS は、すべての CAPSENSE™ ピンの自律スキャンを処理できます。

3.4.6 CAPSENSE™ CSDRM 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。これが耐液性にどのように役立つかの詳細については、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、図 49 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。

[4] 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。

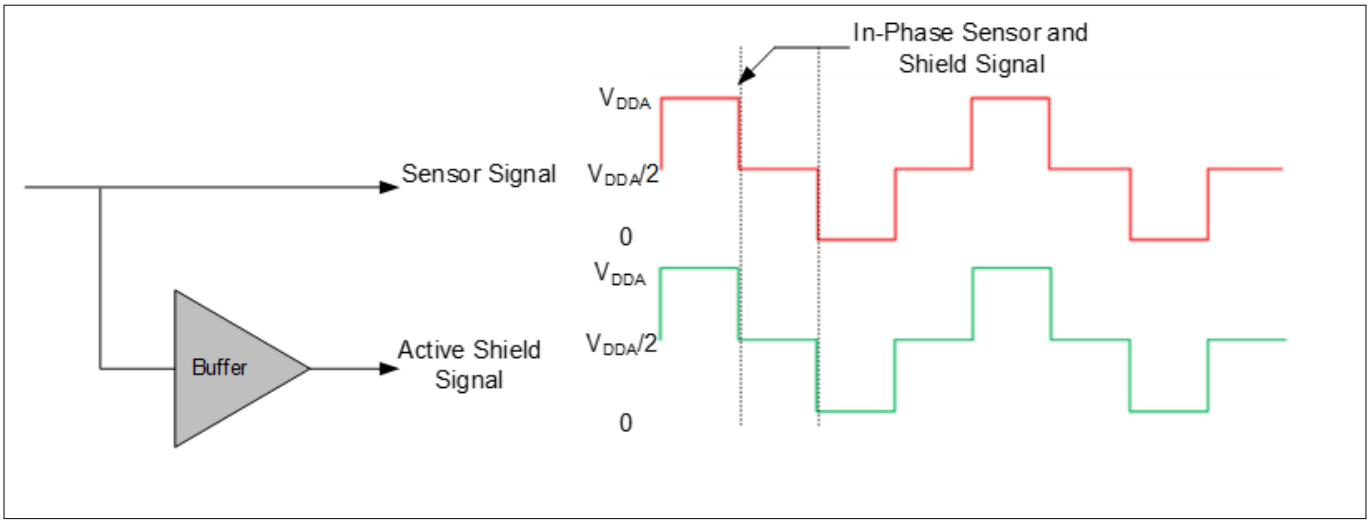


図 49　　アクティブシールド信号

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、図 50 に示すようにシールドが V_{DDA} と GND の間で切り替えられます。切り替えは、センサーとシールド間の正味電荷が 2 つのセンスクロックごとに無効になるように制御されます。

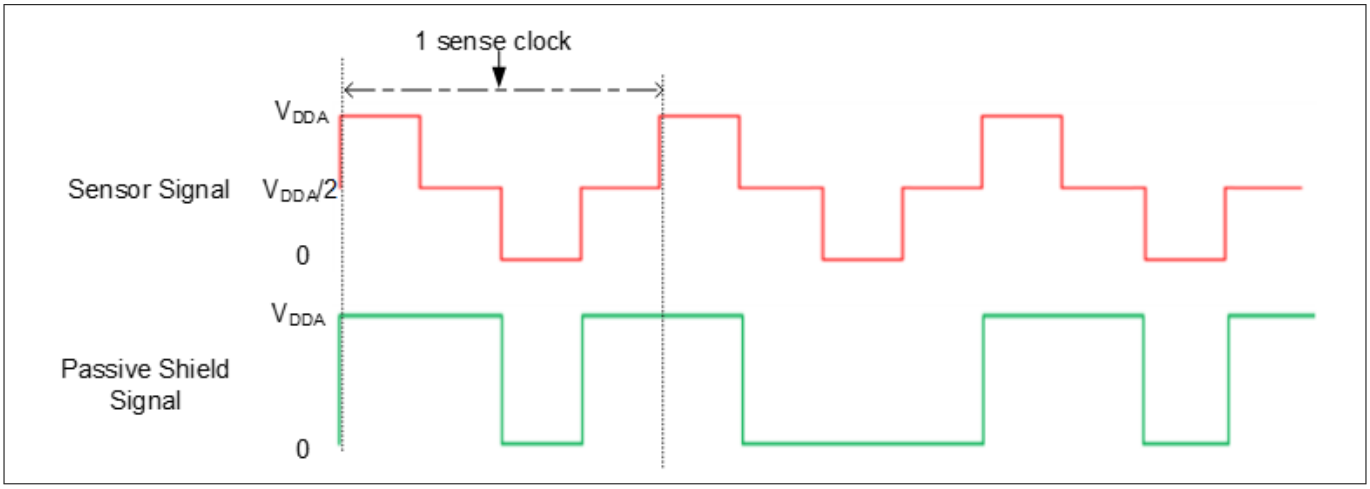


図 50　　パッシブシールド信号

表 5 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。

表 5　　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 世代および第 5 世代低消費電力)

図 51 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})

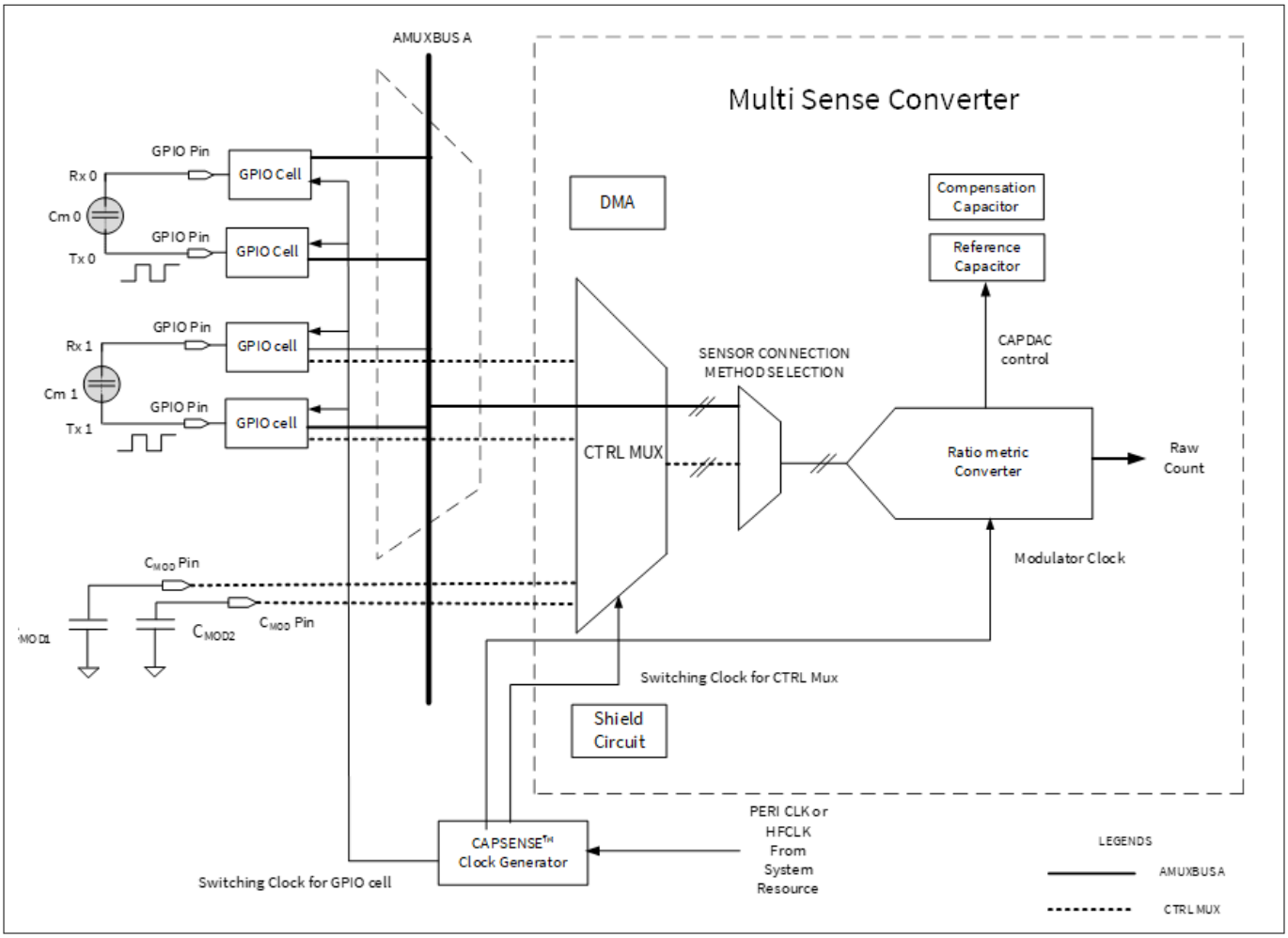


図 51 CAPSENSE™ CSX-RM センシング方式の構成 (第 5 世代)

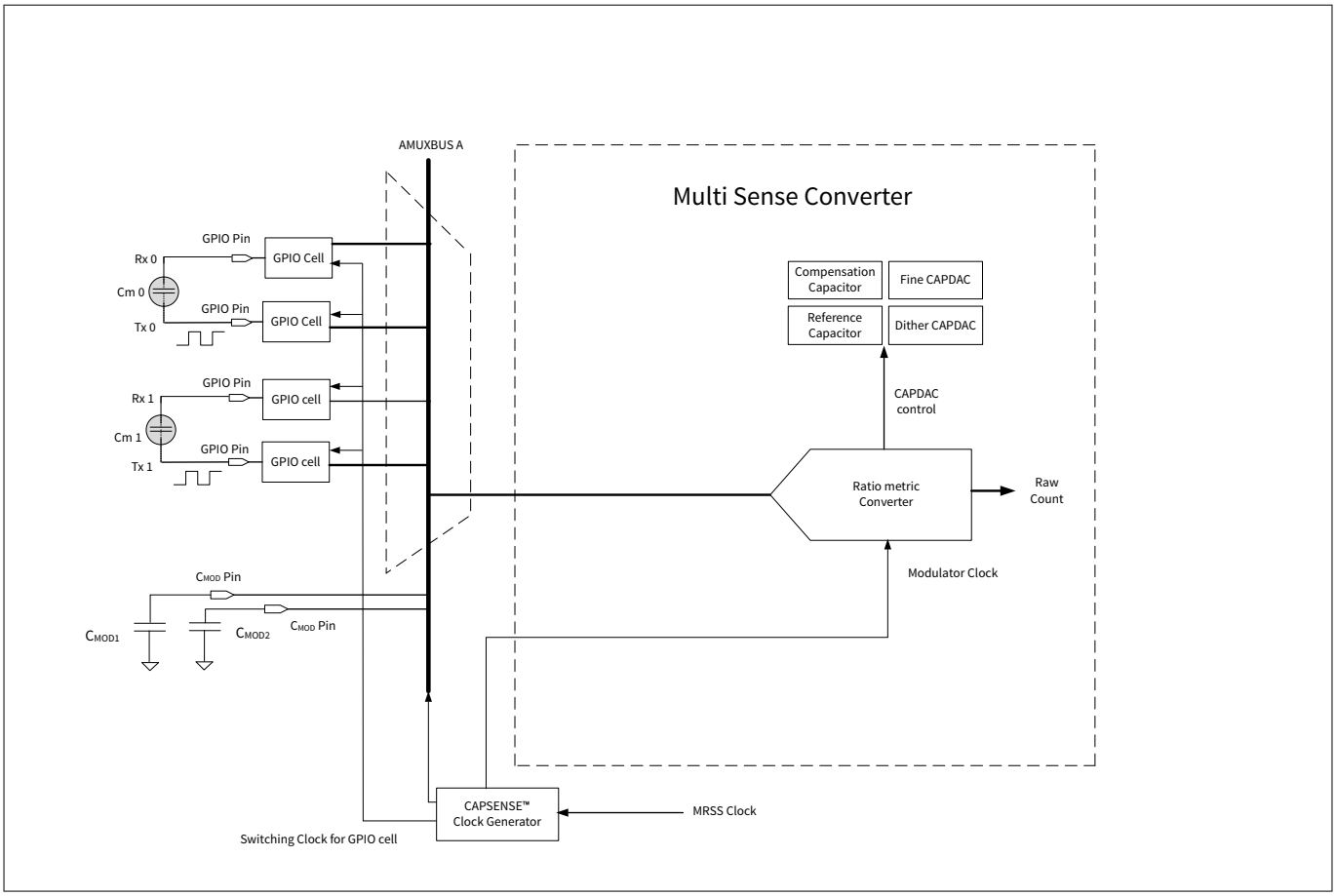


図 52　　CAPSENSE™ CSX-RM センシング方式の構成 (第 5 世代低電力)

CSX-RM センシング方式では、図 51 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、必要に応じて 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。

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 24 に従って Tx クロック分周器の値で表されます。

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

式 24　　Tx クロック分周器

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。

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

式 25　　補償 CDAC 分周器

ここで、

$CompClk_{Div}$ = 補償 CDAC 分周器

FM_{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。

- 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 26 で与えられます。

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

式 26　　CSX-RM シングル CDAC raw カウント

デュアル CDAC モードでは、補償 CDAC は常にオンです。C_{comp} が補償 CDAC の場合、raw カウントの式は式 27 で与えられます。

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

式 27　　CSX-RM デュアル CDAC raw カウント

ここで、

MaxCount = N_{Sub}.$TxClk_{Div}$

N_{Sub} = サブ変換回数

$SnsClk_{Div}$ = 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

CIC2 を有効にすると、

$$\text{MaxCount} = \text{Decimation Rate}^2 * K$$

式 28　　最大 raw カウント

ここで、

$$\text{Decimation rate} = \frac{\text{SnsClk}_{Div} * N_{sub}}{3}$$

式 29　　デシメーションレート

$$K = \frac{\text{No. of CIC2 samples}}{\text{CIC2 Hardware divider}}$$

式 30

可能な限り高い raw カウントを得るために、"K"を "1"に設定できるように、CIC2 のサンプル数を選択することを推奨します。

表 6　　CIC2 ハードウェア分割ルックアップテーブル

CIC2 サンプル数	分周器用推奨 CIC2 シフト	CIC2 ハードウェア分周器
2	1	2
3〜4	2	4
5〜8	3	8
9〜16	4	16
17〜32	5	32
33〜64	6	64
65〜128	7	128
129〜256	8	256

CAPSENSE™コンフィギュレーターで CIC2 シフトの自動オプションを選択すると、表 4 に従ってハードウェアシフト（ハードウェアデバイダ）が選択されます。

デシメーションレートはダウンサンプリングレートであり、最大 255 に達した時、CIC2 アキュムレータ(25 ビット)のオーバーフローを回避するために以下の条件を確保する必要があります。

$$\text{CIC2_Acc_Size}_{Max} \geq \text{TRUNC}\left(\frac{N_{sub} * \text{Sns_Clk_Div}}{\text{Decimation}} - 1\right) * \text{Decimation}^2$$

式 31　　CIC2 アキュムレータのオーバーフロー状態

ここで、

$\text{CIC2_Acc_Size}_{Max}$ - CIC2 アキュムレータの最大サイズ

最大 CIC2 アキュムレータ サイズは、ダイレクト クロックの場合は $(2^{25} - 1)$、PRS クロックの場合は $(2^{25} - 1)/2$ です。

式 27 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 第 5 世代および第 5 世代低消費電力の利点

自律スキャンと CIC2 フィルタの詳細については、AN234231 - Achieving lowest-power capacitive sensing with PSOC™ 4000T を参照してください。

3.6.1 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、後続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を参照してください。

第 5 世代の低消費電力 CAPSENSE™では、自律スキャンが本質的にサポートされており、CPU の介入を必要とせず、CTRLMUX と DMA を必要としません。第 5 世代の CAPSENSE™テクノロジでは、CTRLMUX によって限られたピン数がサポートされていましたが、すべての CAPSENSE™ピンが AMUXBUS 経由で自律スキャンモードをサポートします。

3.6.2 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 世代および第 5 世代低電力)および CAPSENSE™ CSX-RM センシング方式 (第 5 世代および第 5 世代低消費電力)のセクションで説明したものと同じ機能とパフォーマンスを備えます。各インスタンスはチャ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ネルを意味します。マルチチャネルの動作は、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。チャネルの動作は同期され、チャ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ネルノイズを回避します。

詳細については、「マルチチャネルスキャン」セクションを参照してください。

3.6.3 CIC2

マルチセンス ローパワー (MSCLP) CAPSENSE™デバイスは、カスケード積分櫛形 2 (CIC2) デジタル フィルタを内蔵しており、所定のスキャン期間において実効分解能を向上させ、それによって SNR を向上させます。CIC2 フィルタは、デルタ シグマ コンバータのフィルタに使用される 2 次デジタル ローパス (デシメーション) フィルタです。図 53 は、2 つの積分器と 2 つの櫛形フィルタのカスケードで構成される CIC2 フィルタを示します。

CIC2 フィルタは、デルタ シグマ コンバータである MSCLP アナログ フロントエンドの出力を受信します。このコンバータは、入力を表す 1 と 0 のビットストリームを生成し、量子化ノイズを高周波に移動させます。この高周波ノイズはデジタル ローパス フィルタでフィルタされ、ダウンサンプラは入力を測定信号を表す 1 つのデジタル ワードに変換します。このローパス フィルタ (A0、A1) とダウンサンプラ (D0、D1) の組み合わせは、図 53 に示すように、デシメータ (CIC2 フィルタ) として知られています。

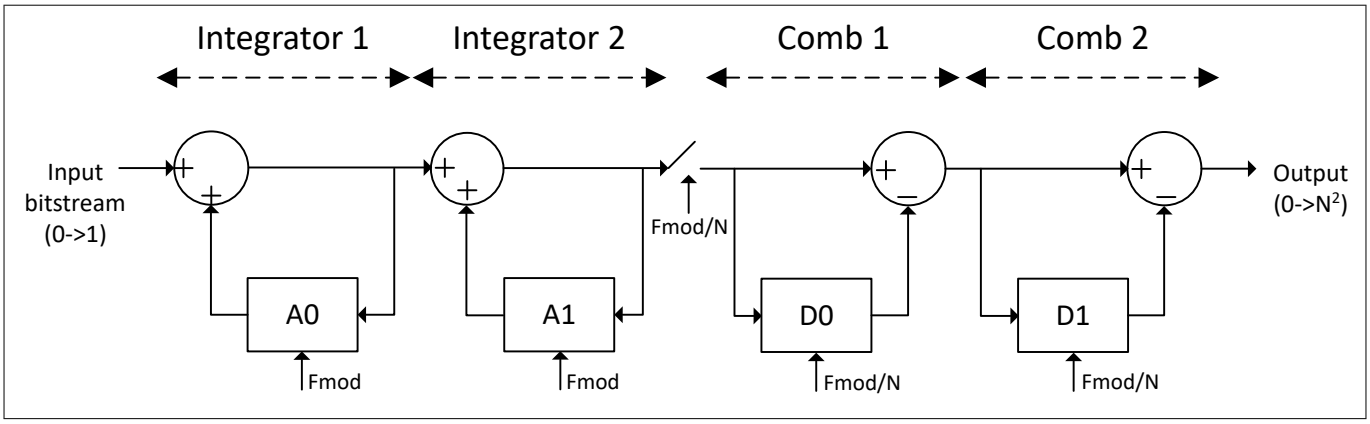


図 53　　CIC2 フィルタ ブロック ダイヤグラム

raw カウントは、各有効サンプルの終了時に累積されます。

適切な CIC2 フィルタには、最低 2 つの有効な CIC2 サンプルが必要です。2 つの有効なサンプルを考慮し、以下の式を用いてデシメーション (ダウンサンプリング) レートを計算してください:

[5]

$$\text{Decimation Rate(N)} = \frac{\text{SnsClk}_{Div} * N_{sub}}{3}$$

式 32　　デシメーションレート

CAPSENSE™ コンフィギュレータで、CIC2 シフトパラメータを "Auto "に設定してください。これにより、適切なハードウェア シフト (ハードウェア分周器) が自動的に選択されます。

CIC2 が有効な場合、以下の式で最大生カウントを計算してください:

$$Rawcount_{max} = \text{Decimation Rate}^2$$

式 33　　最大 raw カウント式

5　デシメーションレートが最大値である 255 に達した場合、CIC2 アキュムレータ（25 ビット）のオーバーフローを回避するために、ユーザは以下の条件が満たされていることを確認する必要があります。

ダイレクトクロック: $(2^{25}\text{-}1) >= \text{TRUNC}((Nsub*Sns_Clk_Div)/Decimation\text{-}1)*Decimation^2$

PRS クロック: $(2^{25}\text{-}1)/2 >= \text{TRUNC}((Nsub*Sns_Clk_Div)/Decimation\text{-}1)*Decimation^2$

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します。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。表 7 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。

表 7 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™ 4000T, PSOC™ 4100S Max, 全 PSOC™ 6 デバイス	ModusToolbox™	CAPSENSE™ ミドルウェア
他のすべての PSOC™ 4 デバイス	PSOC™ Creator	CAPSENSE™コンポーネント

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。詳細については 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)。

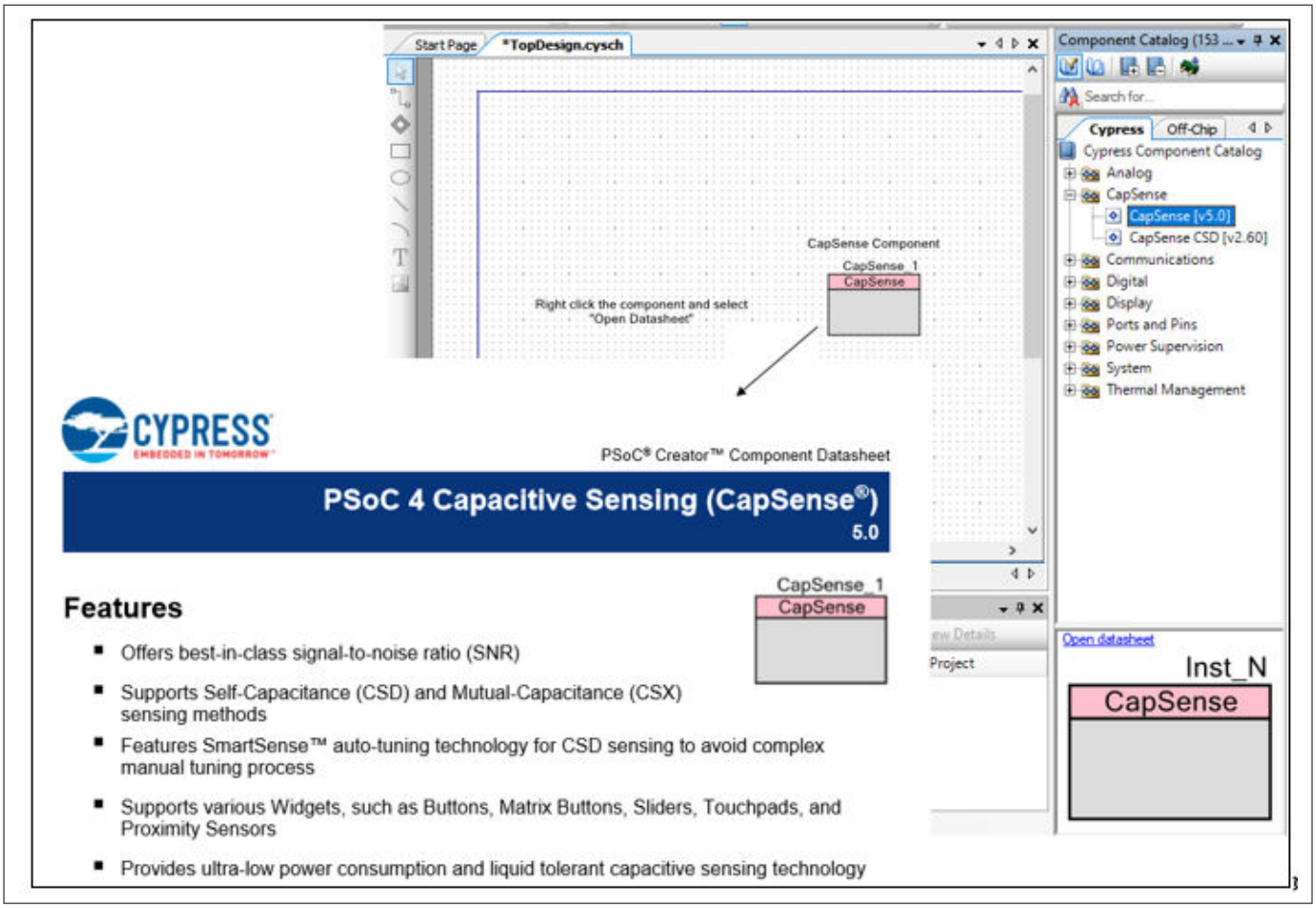


図 54　　PSOC™ Creator コンポーネントの配置

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_ADC コンポーネント

CapSense_ADC[6]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 の使用手順の詳細については該当コンポーネント データシート/ミドルウェア ドキュメントを参照してください。

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。

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、図 55 に示すように、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。

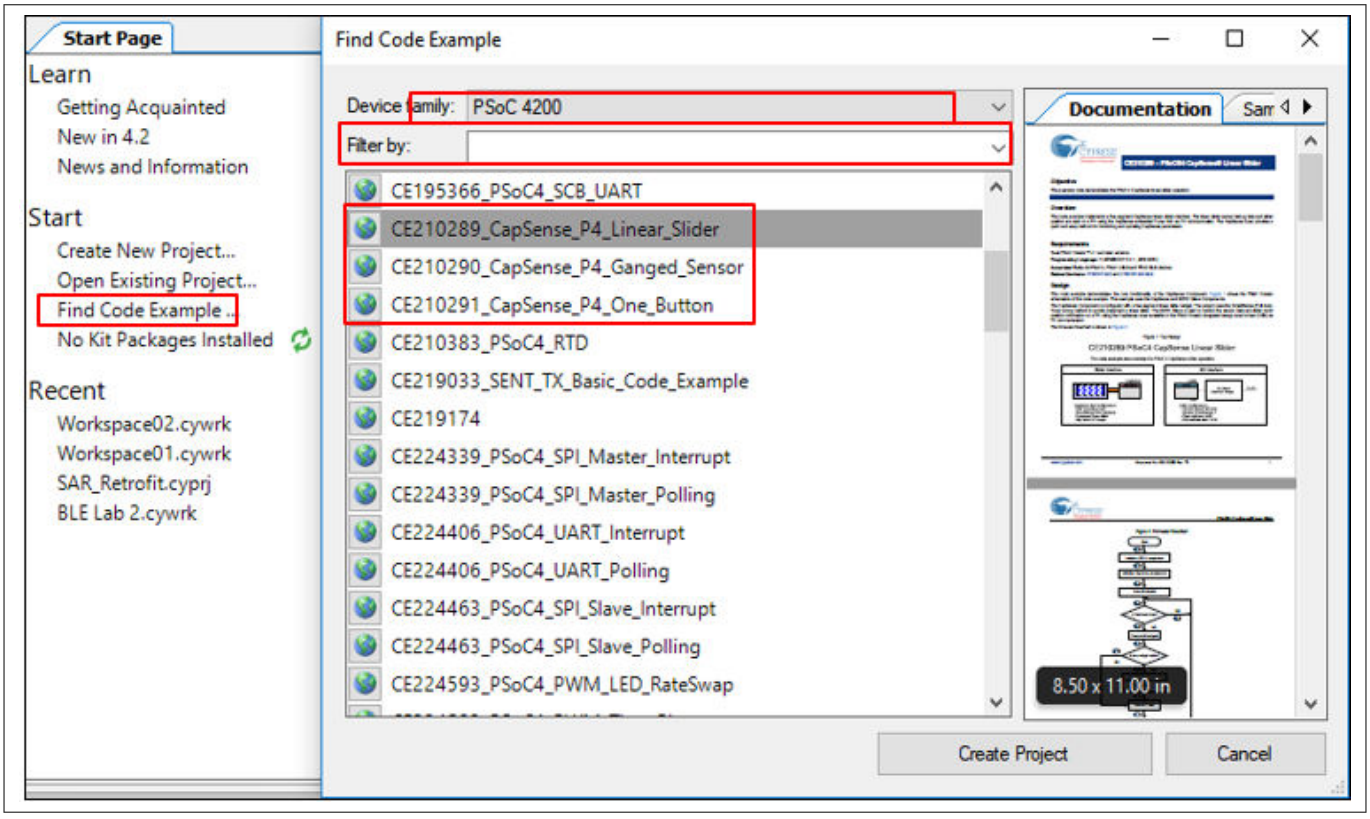


図 55 PSOC™ Creator のサンプル プロジェクト

4.2 ModusToolbox™

PSOC™ 6 および PSOC™ 4[7] 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™ソフトウェアのトレーニングは、こちらを参照してください。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ーを構成するだけで、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。

7 ModusToolbox™でサポートされている PSOC™ 4 デバイスについては、表 7 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 パラメーター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。図 56 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。

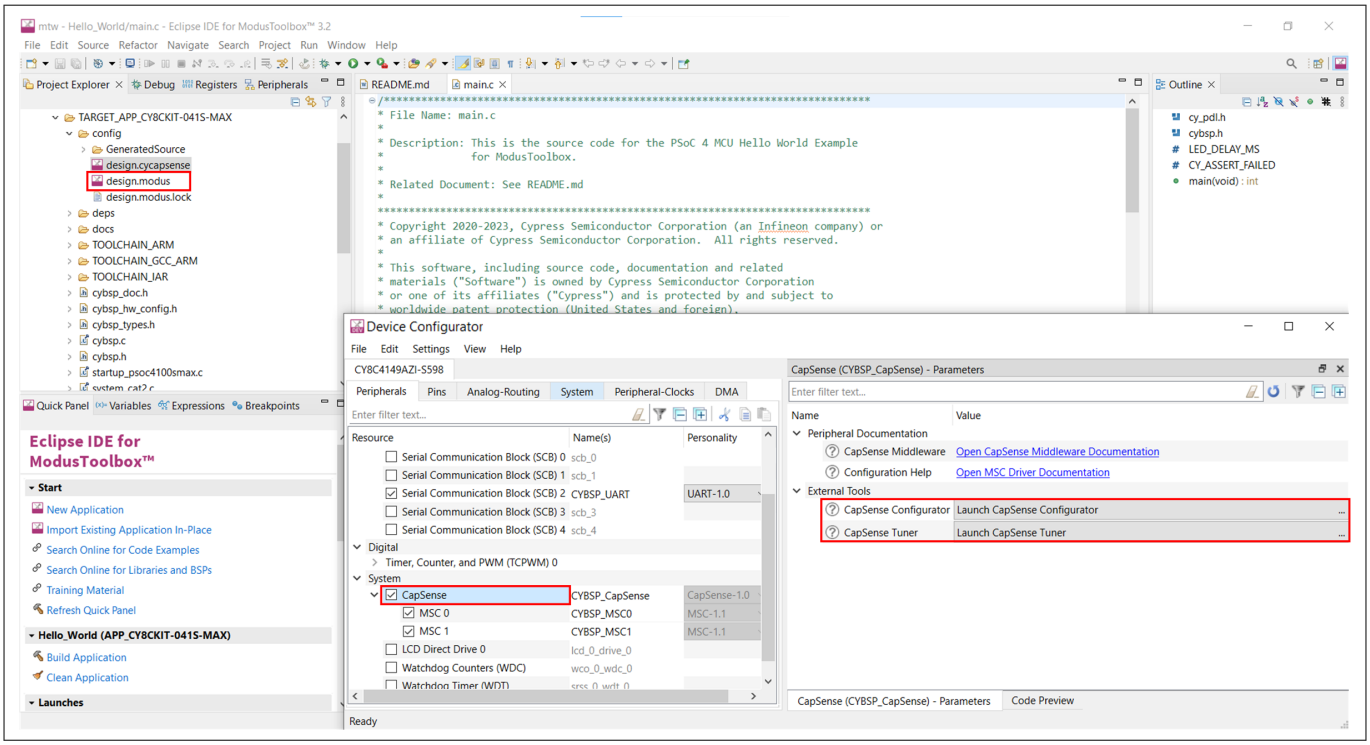


図 56 ModusToolbox™での CAPSENSE™構成ツール

4.2.3 CSDADC ミドルウェア

この CSDADC ミドルウェア [8]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、図 56 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。

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。

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。図 57 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。

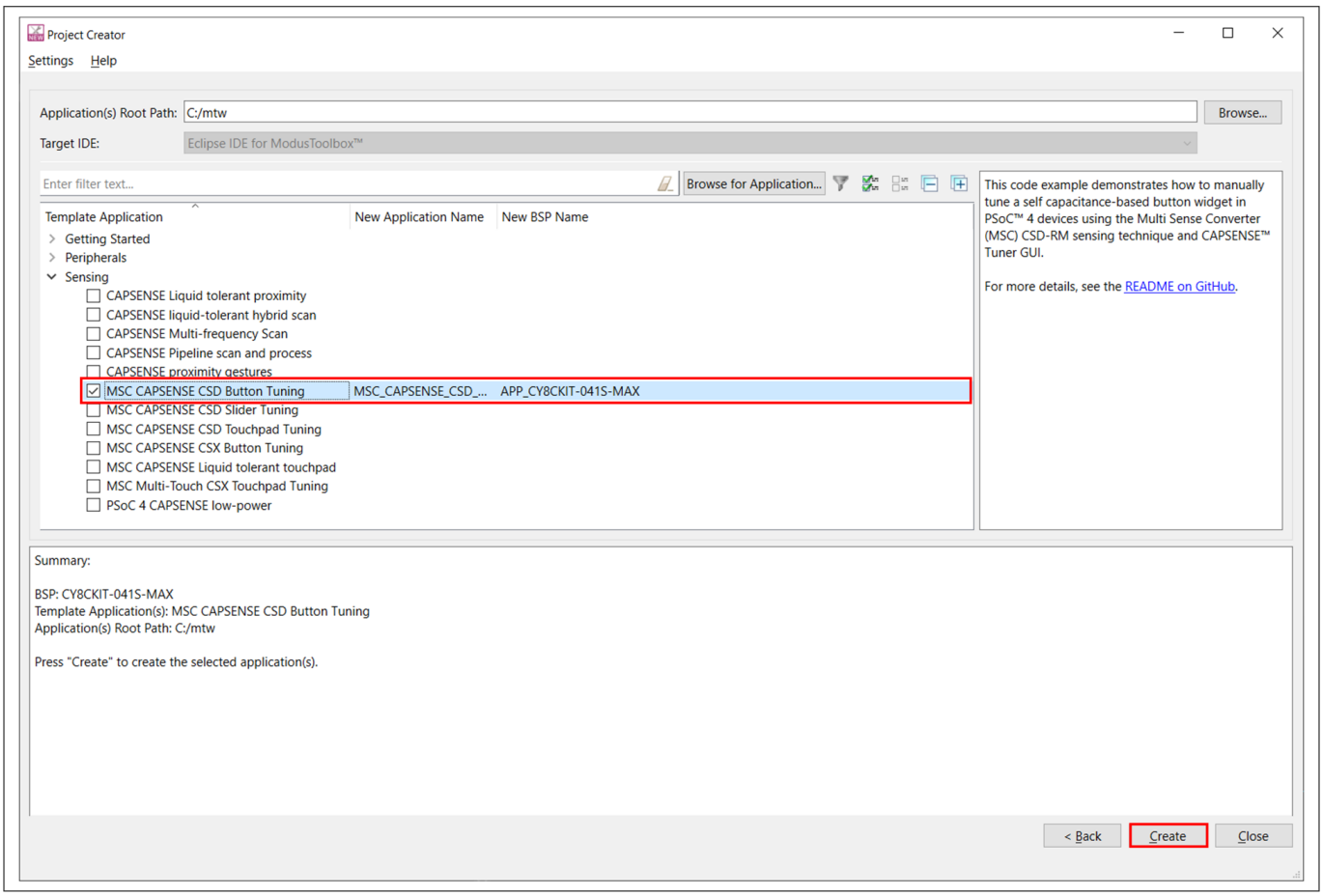


図 57 ModusToolbox™での CAPSENSE™ CSD ボタンサンプルプロジェクトの作成

4.3 ハードウェア キット

表 8 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。

表 8 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 7×7 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

(続く)

表 8 (続き) 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™ 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 1 個の 8 素子の自己容量または相互容量センシングリニア スライダー 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™ 4000T Kit (CY8CKIT-040T)	1 個の CAPSENSE™ボタン 1 個の 4x5 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™ 4000T プロトタイピング キット (CY8CPROTO-040T)	1 個の自己容量センシングボタン 1 個の相互容量センシングボタン 1 個の近接センサーループ 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™ 4 Pioneer Kit (CY8CKIT042)	1 個の 5 素子リニア スライダー
PSOC™ 4 Bluetooth® LE Pioneer Kit(CY8CKIT-042-BLE)	1 個の 5 素子リニア スライダーおよび 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 スライダー
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 (CY8CKIT038)、PSOC™開発キット (CY8CKIT001)	1 個の 5 素子リニア スライダーおよび 2 個のボタン
CAPSENSE™拡張基板キット CY8CKIT-031 (CY8CKIT038 および CY8CKIT-001 に使用)	1 個の 10 素子スライダー、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

(続く)

表 8 (続き) 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™ 6 Wi-Fi BT pioneer kit (CY8CKIT-062-WiFi-BT pioneer kit) および PSOC™ 6 Bluetooth® LE Pioneer Kit (CY8CKIT062-BLE pioneer Kit)	1 個の 5 素子 CAPSENSE™スライダー、2 個の CAPSENSE™ ボタン、1 個の CAPSENSE™近接センシング ヘッダ、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™スライダーおよび 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 ドキュメントまたは 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 ドキュメントまたは 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。

***注**: 手動チューニングは、I²C または UART とホスト PC との通信を必要とします。*

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。

注: *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します。**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。図 58 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。

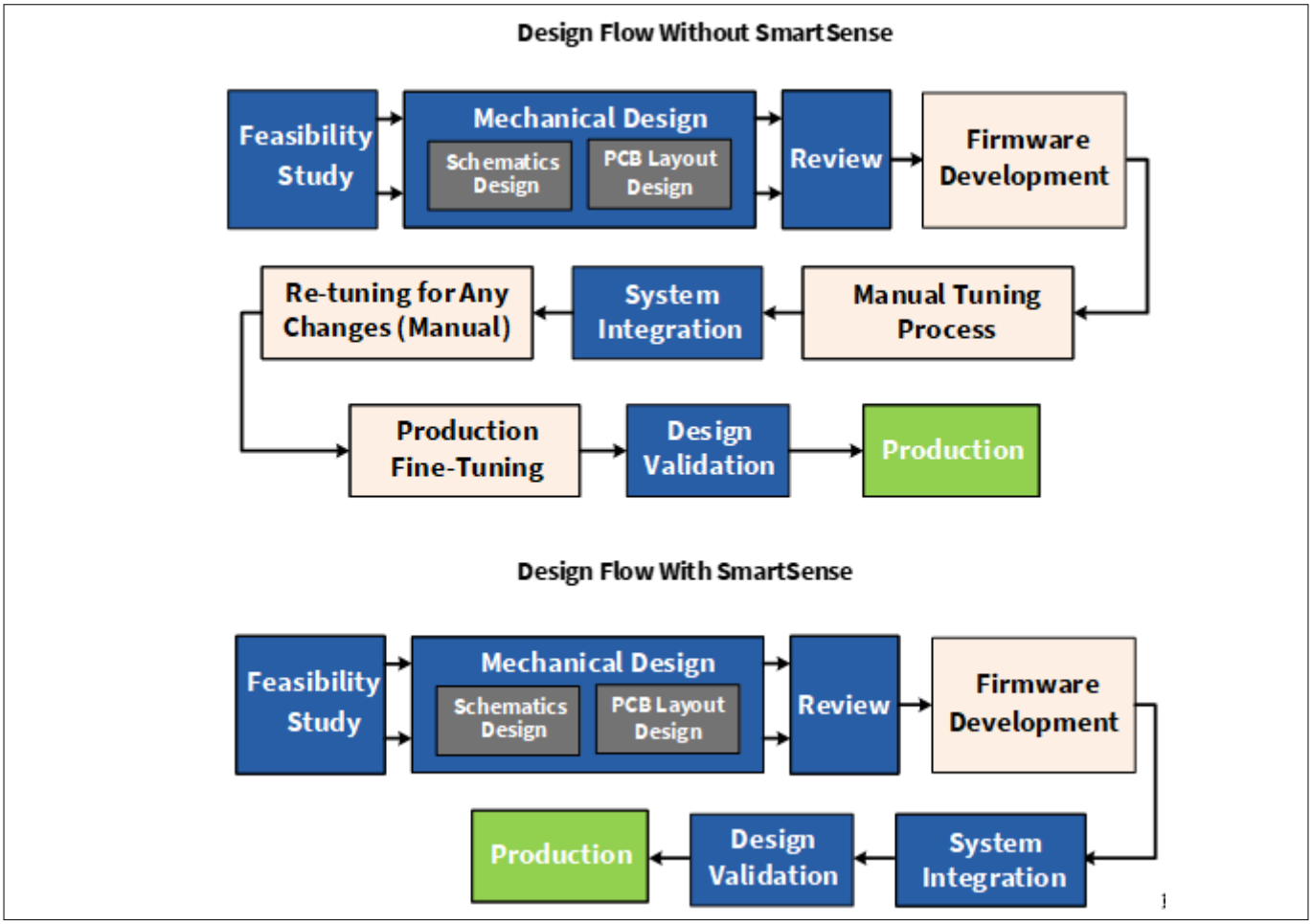


図 58 SmartSense を使用する場合と使用しない場合の設計フロー

- **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。
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 (表 14 を参照) を設定できます。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。

4.

表 9 SmartSense で自動チューニングされる CAPSENSE™パラメータ

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		
変調器クロック周波数		
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。表 14 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ーウィジェットで異なります。

***注**: 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。

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ー値を減らしてください。図 59 に、設計で指の静電容量パラメーター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、表 10 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。

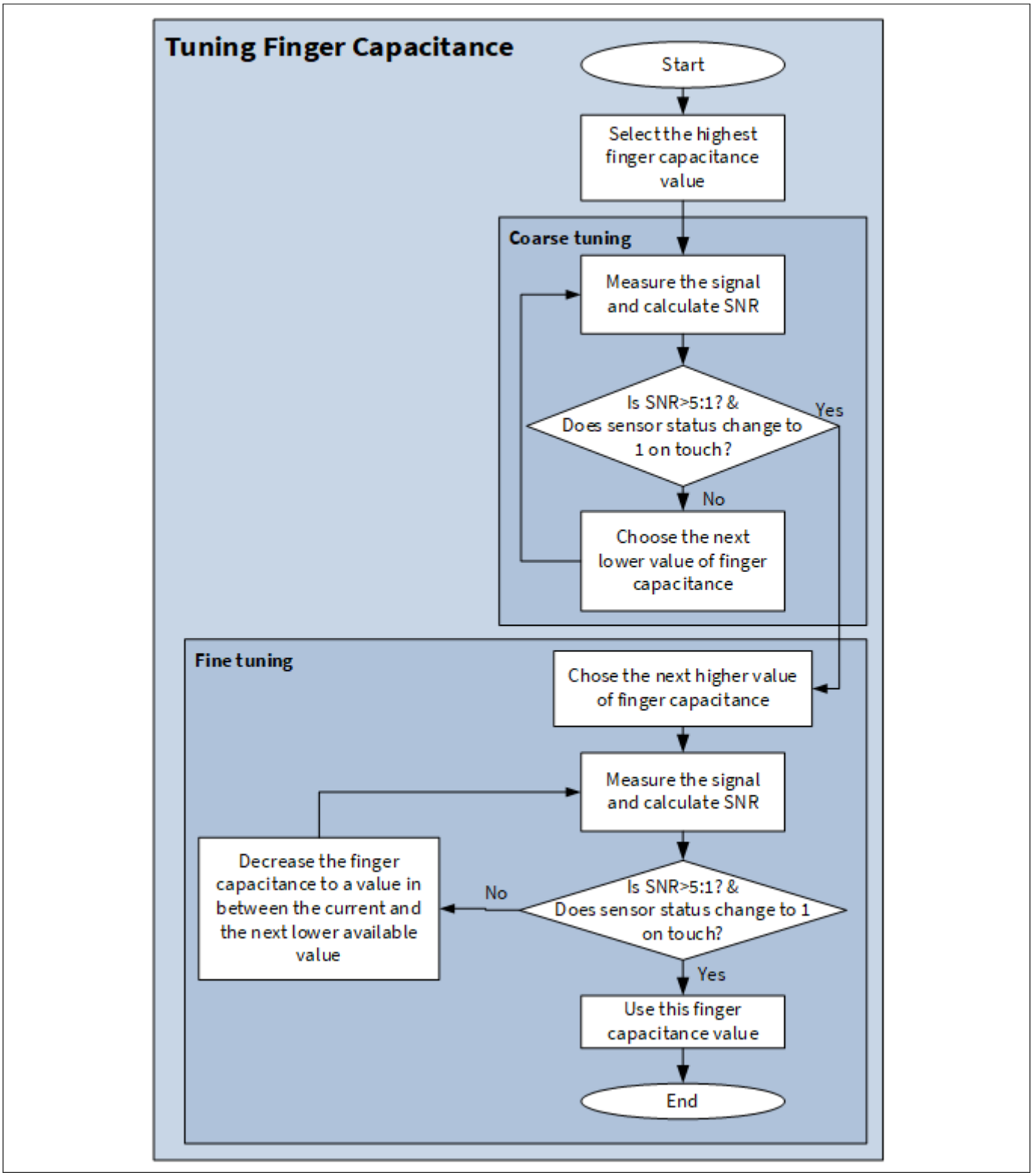


図 59　　PSOC™ Creator での SmartSense 自動調整ベースの CAPSENSE™プロジェクトの使用

表 10 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。	$y[i] = \frac{1}{4} \times (x[i] + x[i-1] + x[i-2] + x[i-3])$	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。	$y[i] = \frac{1}{K} \times \{N \times x[i] + (K-N) \times y[i-1]\}$	高周波ノイズ除去
ハードウェア IIR	通常のウィジェット (近接と低消費電力を除く) のハードウェア IIR フィルタ (右の式を参照) を有効にします。 iirRCcoef - IIR フィルタの raw カウント係数; 有効範囲: 0～8。係数が低いとフィルタが甘くなり、係数が高いと応答時間が長くなります。 注: 係数値 "0 "に対するフィルタはありません。	$RawCount = \frac{1}{2^{iirRCcoef}} RawCount_{New} + \left(1 - \frac{1}{2^{iirRCcoef}}\right) RawCount_{Previous}$	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。図 60 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、スライダーの設計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。図 60 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。

注: 補正 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。

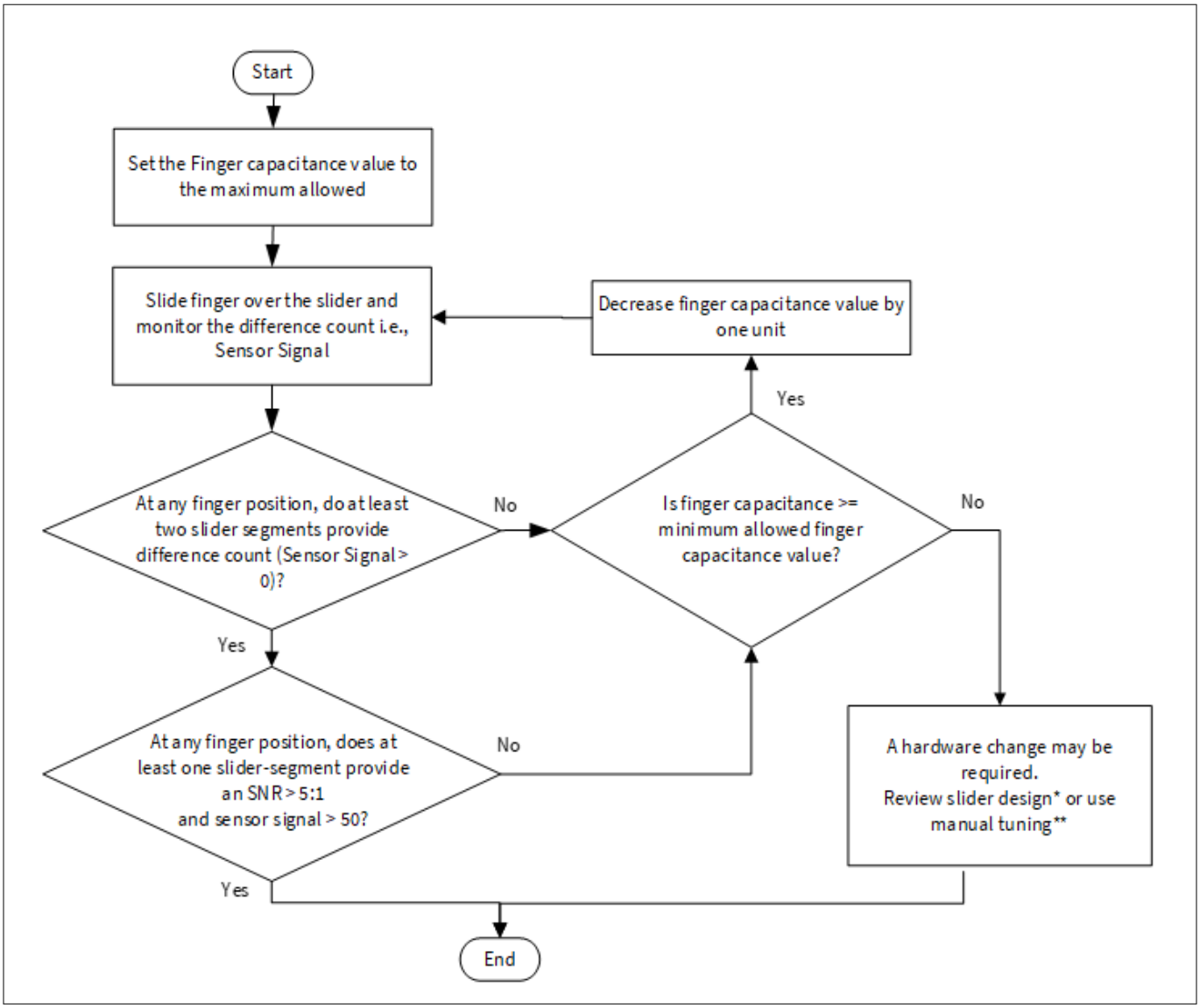


図 60　　スライダーの指容量値の設定

* 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 - CAPSENSE™による近接センシングおよび CAPSENSE™入門の「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、表 14 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

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 ドキュメントを参照してください。

図 61 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。

SmartSense を使用して CAPSENSE™ハードウェア パラメーターの選択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。

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。

CAPSENSE™ソフトウェア パラメーターの選択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。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表 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。図 61 に一般的な手動チューニング手順を示します。



図 61　　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。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。

注: 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ります。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、表 10 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。

$$\text{rawcount} = G_{CSD}\, C_S$$

式 34 raw カウントとセンサー静電容量の関係

ここで、

C_S = センサーの静電容量

C_S = センサー上に指がないときの C_P

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 10 および式 34 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。

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

式 35 静電容量-デジタル変換利得

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

式 36 静電容量-デジタル変換利得 (IDAC シンクモード)

ここで、

V_{REF} = コンパレータの基準電圧。表 2 を参照してください。

F_{SW} = センス クロック周波数

I_{MOD} = 変調 IDAC 電流

N = シグマ-デルタ変換器の分解能

変換利得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。図 62 に、raw カウントとセンサー容量のプロットを示します。

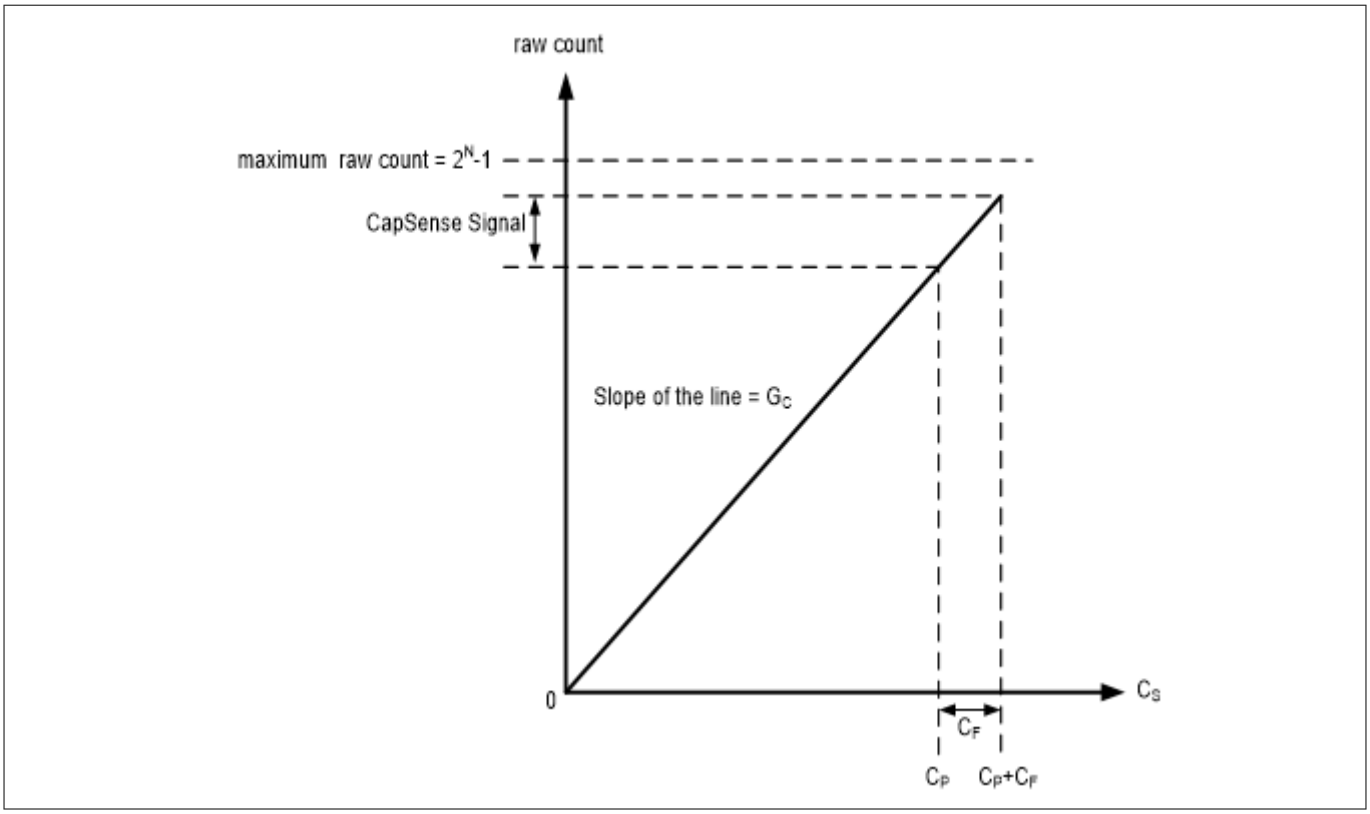


図 62　　センサー容量に対する raw カウント

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。図 63 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。

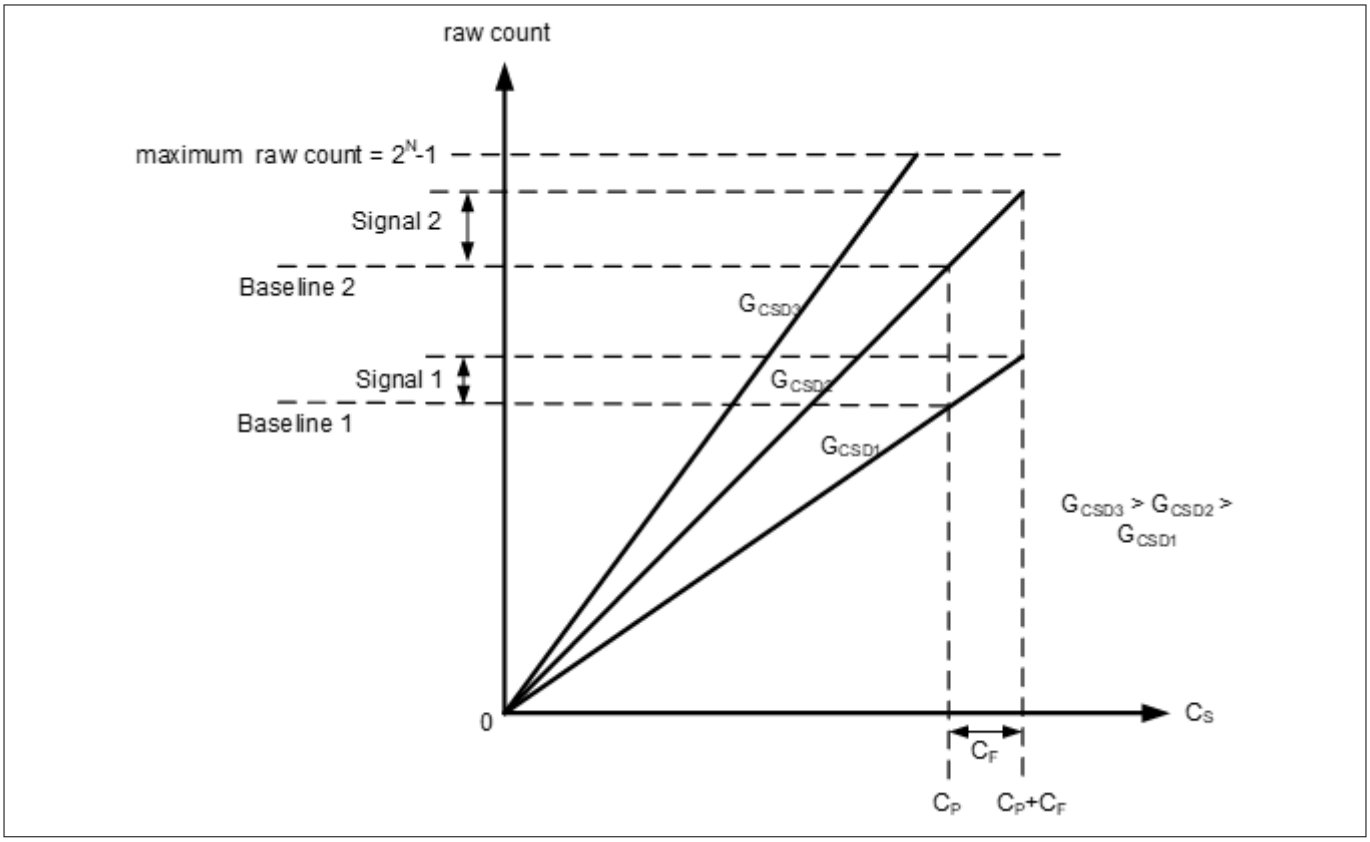


図 63　　さまざまな変換利得の信号値

図 63 に、3 つの変換利得値 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 (2^N-1) 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。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。図 64 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= (2^N-1)の 85%になるように CSD パラメーターを調整します。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。

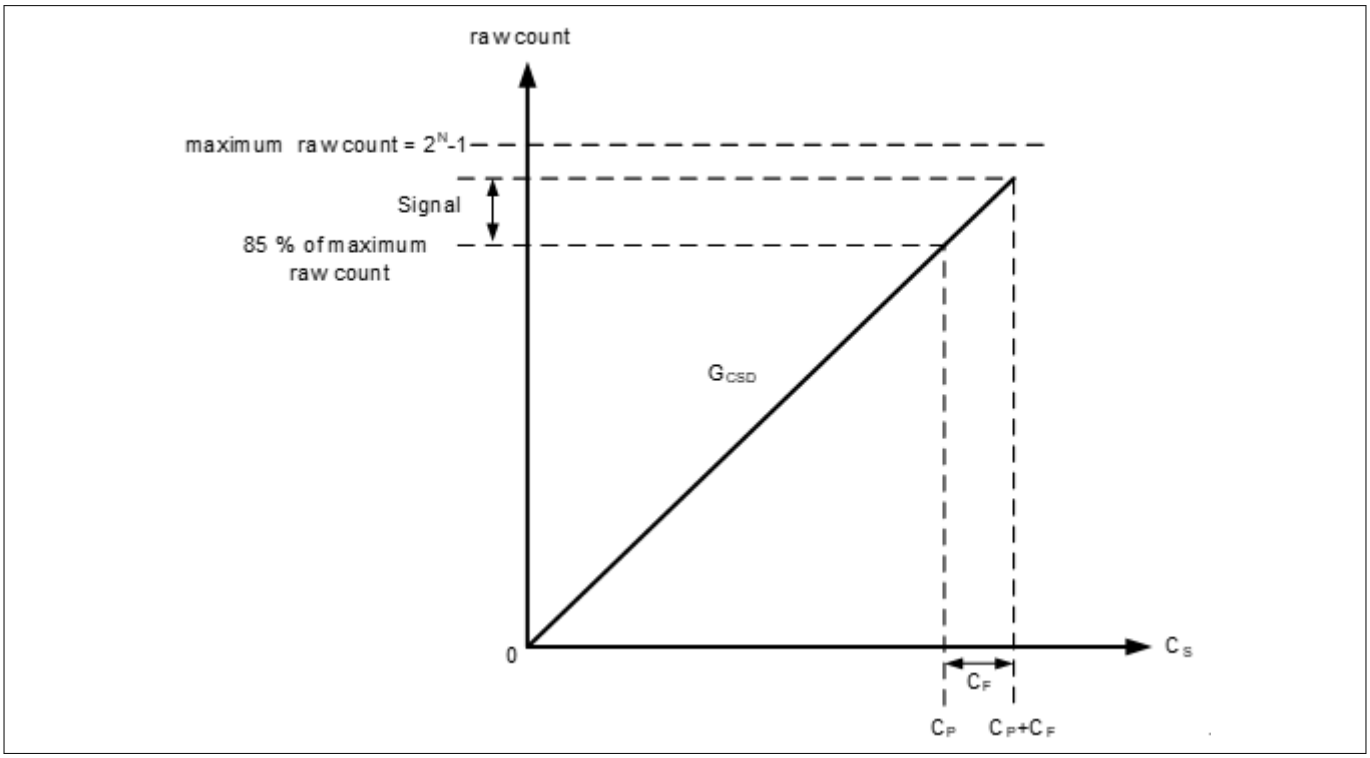


図 64　　推奨チューニング

デュアル IDAC モードでの変換利得

式 34 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。

$$\text{rawcount} = G_{CSD}C_S - \left(2^N - 1\right)\frac{I_{COMP}}{I_{MOD}}$$

式 37　　デュアル IDAC モードでの raw カウント

ここで、

I_{COMP} =補正 IDAC 電流

G_{CSD} は、IDAC ソースモードの場合は式 9 で、IDAC シンクモードの場合は式 36 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、図 65 に示すように、raw カウント= (2N-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。

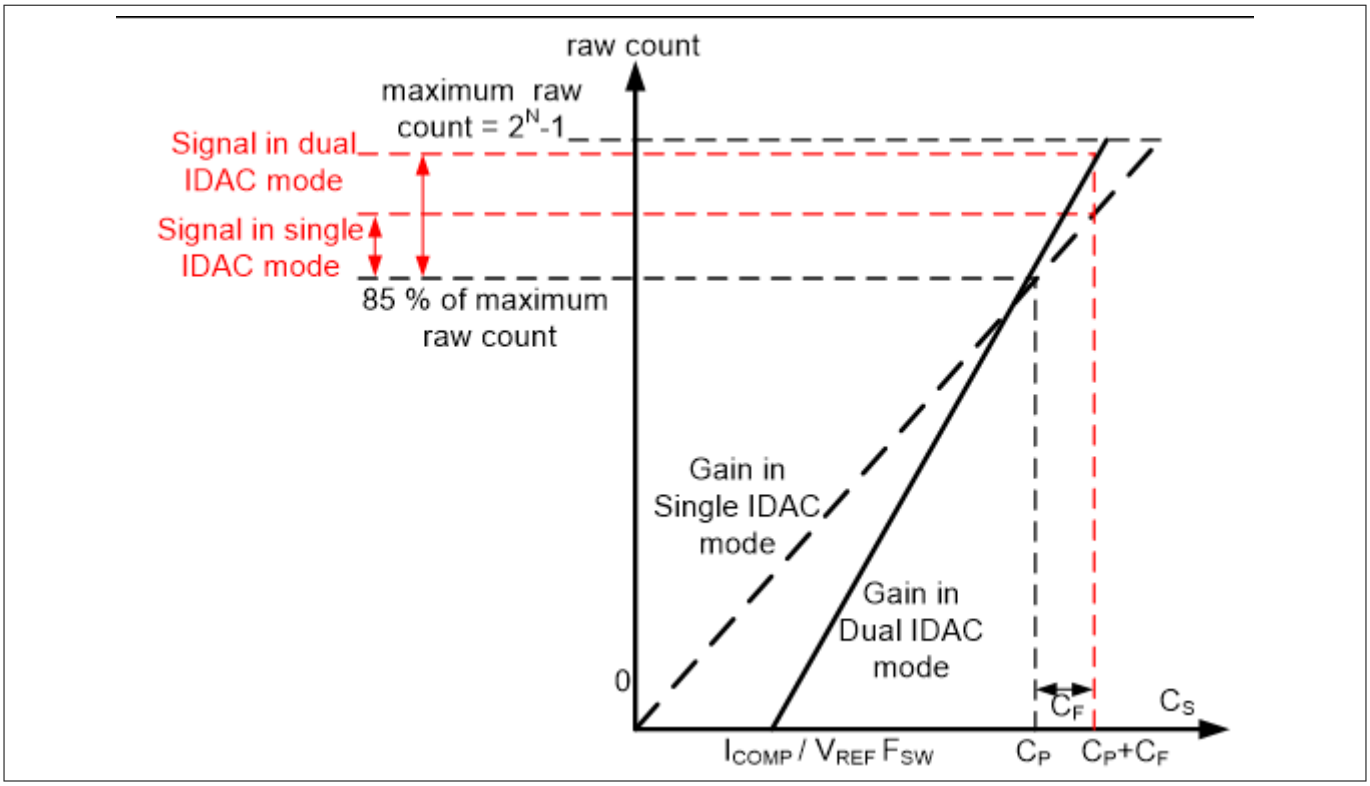


図 65　　デュアル IDAC モードでの推奨されるチューニング

図 65 に示すように、85%の要件は、シングル IDAC モードでは固定利得に制限されますが、デュアル IDAC モードでは、C_S 軸切片を右に移動し (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 カウントと C_P のグラフに次の変更が加えられます。

1. 補償 IDAC を使用すると、以下に示すように、C_S 軸にゼロ以外の切片が導入されます。

$$C_S \text{axis intercept} = \left(\frac{I_{COMP}}{V_{REF} F_{SW}}\right)$$

式 38　　I_{COMP} に関する C_S 軸のインターセプト

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 (他のすべてのパラメーターは同じままです) 、デュアル IDAC モードの利得 G_{CSD} は、式 17 によりシングル IDAC モードの利得の 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 26 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、システム全体のパフォーマンスが向上します。しかし、利得 G_{CSD3} のプロットで図 63 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。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。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。式 26 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. 補正 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、いつも補正 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。図 64 および図 65 に示すように、IDAC を調整して最高利得を達成しますが、C_P に対応する raw カウントに、温度変化などの環境の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。
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

理想的には、図 62 と図 65 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。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、ここでは図 66 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。

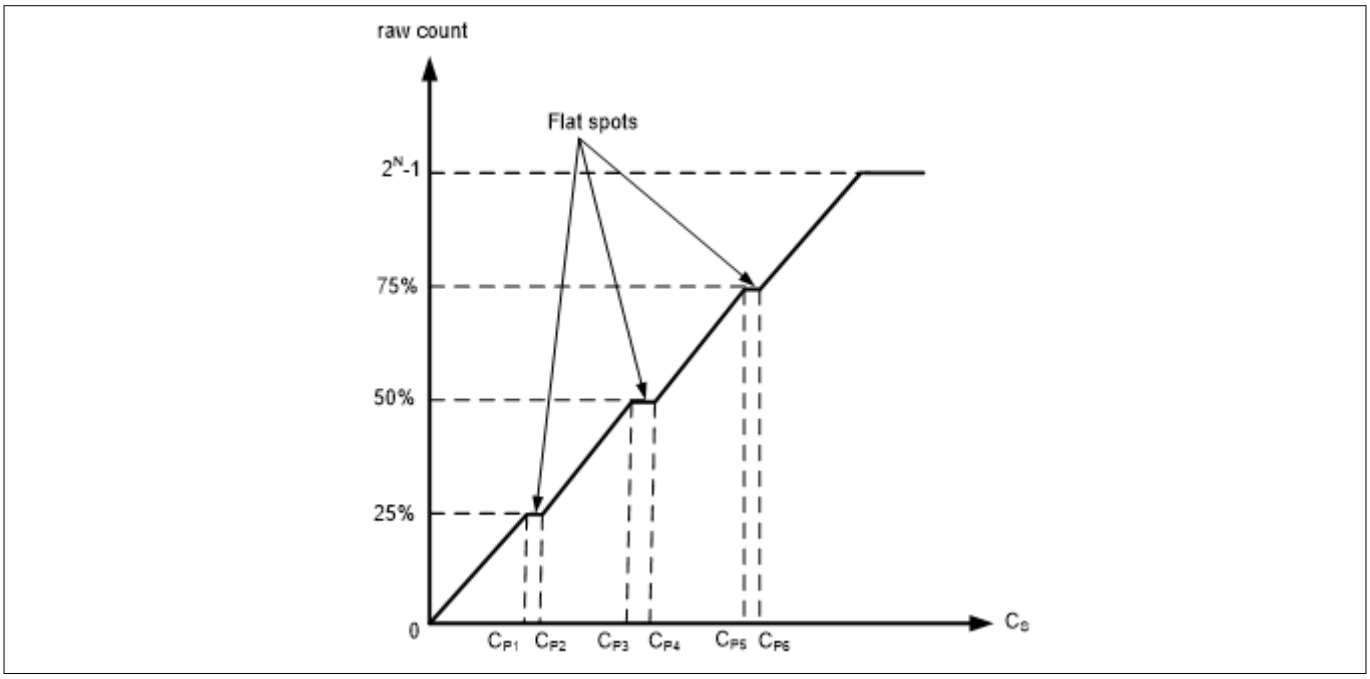


図 66 直接クロック使用時の raw カウント対センサー静電容量のフラット スポット

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。

フラットスポット削減技術

1. 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、2N-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。

2. PRS クロックの使用

   これらのフラットスポットは、直接クロックがセンス 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)。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、センスクロックソースに示されている EMI/EMC の利点も提供するためです。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、変換利得と CAPSENSE™信号を制御します。表 11 に CSD センシング方式に適応する 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器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™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 ドキュメントを参照してください。

表 11 CAPSENSE™コンポーネント ハードウェア パラメーター

Sl.No.	PSOC™ Creator の CAPSENSE™パラメーター	ModusToolbox™での 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	変調器クロック周波数	変調器クロック分周器
4	変調 IDAC	変調 IDAC
5	補正 IDAC	補正 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

表 11 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。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てください。チューナー 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 ドキュメントを参照してください。

図 67 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。

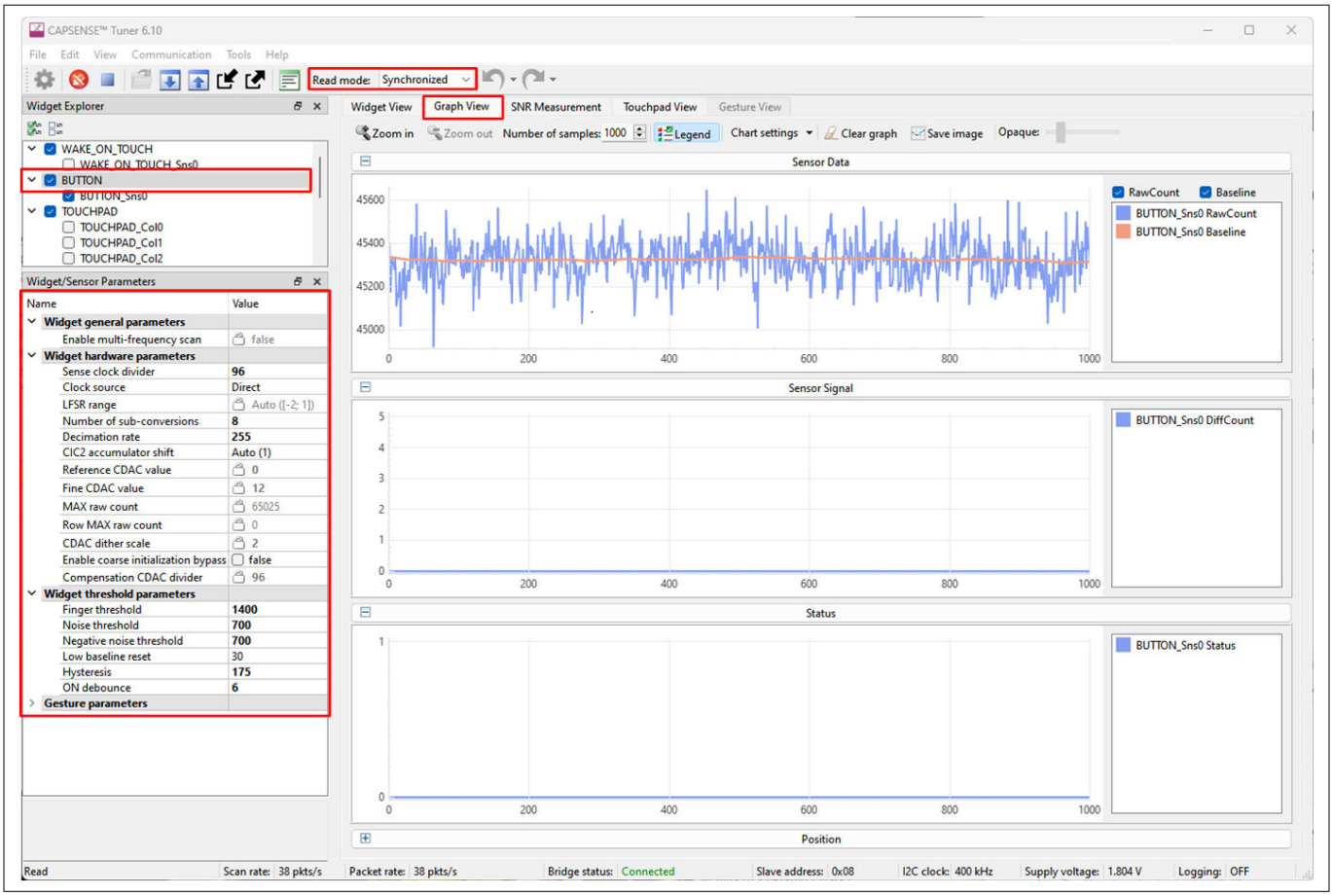


図 67　　チューナー GUI でのハードウェア パラメーター値の読み戻し

5.3.2.2.2　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直接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、$T_{PRS} << T_{SCAN}$ 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。

$$T_{SCAN} = \frac{2^N - 1}{F_{MOD}}, \ where\ N\ is\ the\ Scan\ resolution$$

式 39　　センサー スキャン時間

$$T_{PRS} = \frac{2^{N_PRS} - 1}{F_{SW}}, \ where\ N_PRS\ \ is\ either\ \ 8\ or\ 12$$

式 40　　PRS シーケンス期間

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 ドキュメントを参照してください。

センス クロック周波数

図 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 41 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。

$$F_{SW}\left(\mathrm{maximum}\right) = \frac{1}{10 R_{SeriesTotal} C_P}$$

式 41　　センスクロック最大周波数

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

式 42　　合計直列抵抗

ここで、C_P はセンサーの寄生容量です。そして $R_{SeriesTotal}$ は、内部スイッチの 500 Ω の内部スイッチの抵抗、560 Ω 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 1.06 kΩ にトレース抵抗を加えたものです。

CP 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 ドキュメントを参照してください。

式 35 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、図 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。良好な耐液性を得るには、シールド信号が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 26、式 24 を参照)

- シグマデルタ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ります
- フラット スポットが広くなります

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てください。

変調 IDAC および補正 IDAC

CSD は、図 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

補正 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 モードで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、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 `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、非常に低いまたは高い IDAC コードを回避す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 34, 式 35, および式 37 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 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 39 に従ってセンサーのスキャン時間、式 34, 式 35, および式 37 に従って変換利得を制御します。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 39 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。

注: 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ります。

5.3.2.2.3 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 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ー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、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、シールド波形の整定時間が長くなり、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が落ち着く必要があり、V_{REF} 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。

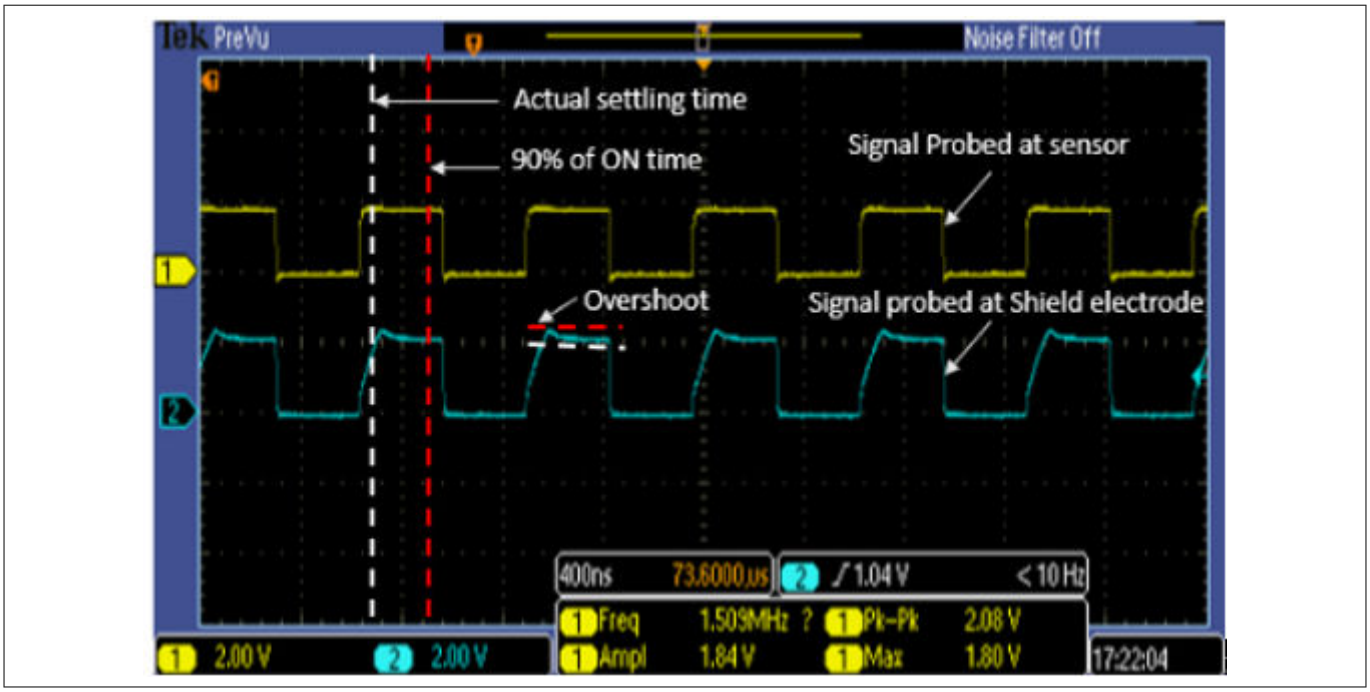


図 68　　適切に調整されたシールド波形

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できます。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。表 12 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。

表 12 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。表 13 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。表 14 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。

表 13 CAPSENSE™コンポーネント ウィジェットしきい値パラメーター

Sl.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

表 14 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。

図 69 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。

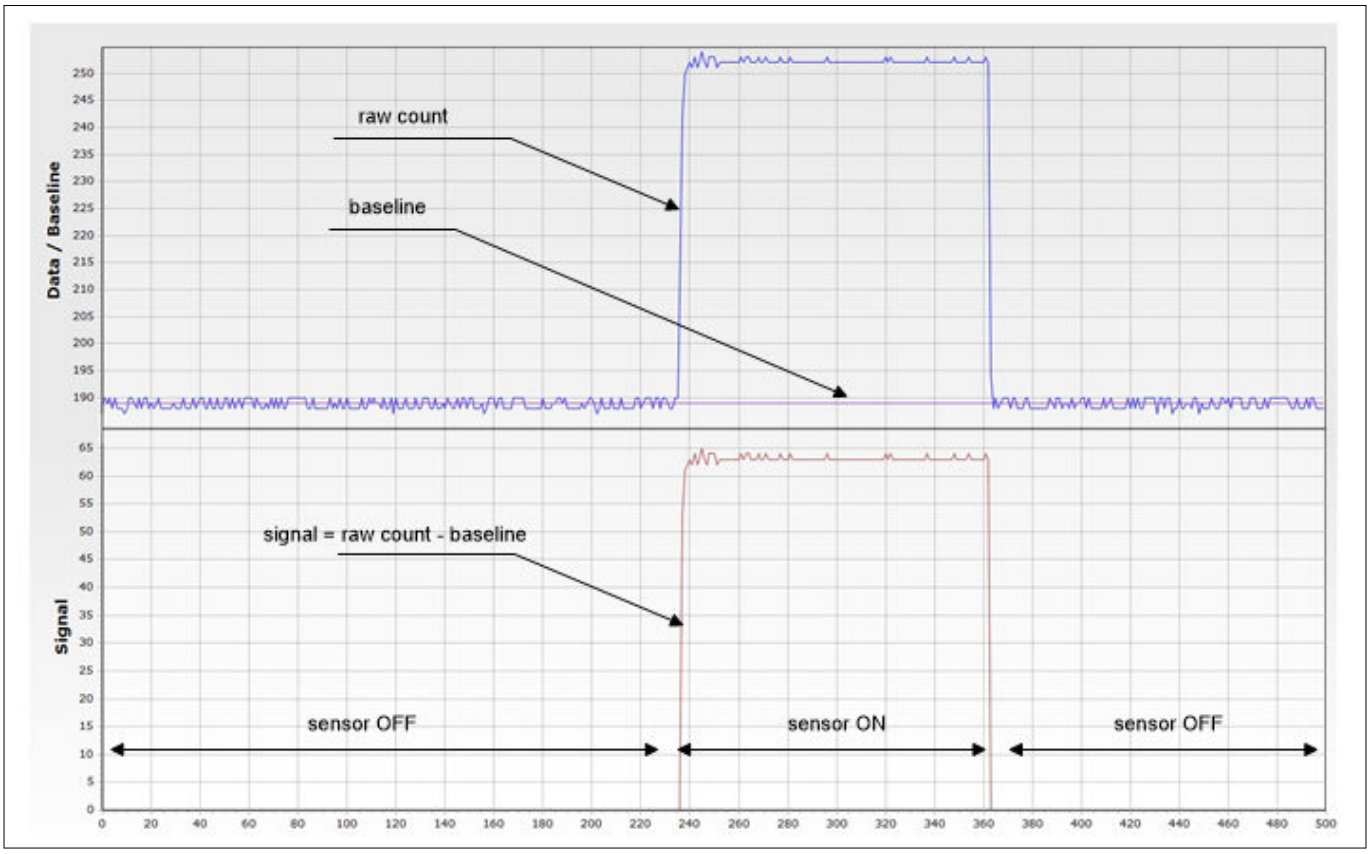


図 69 raw カウント、ベースライン、およびシグナル

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。

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。図 70 に示すように、現在の raw カウントが (*Baseline* – Negative noise threshold) ～ (*Baseline* + 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。

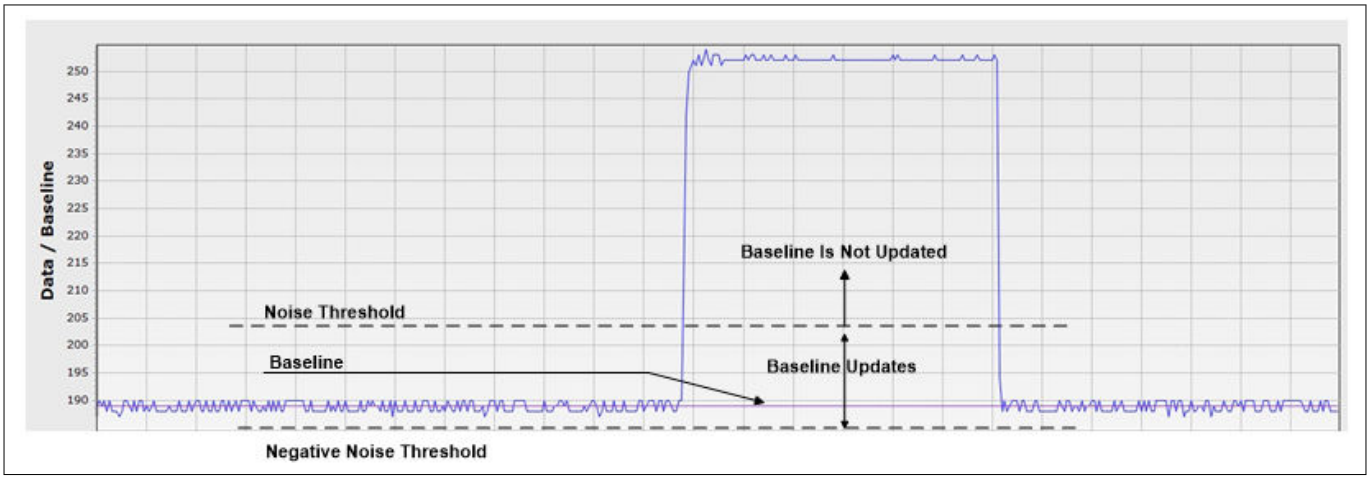


図 70 ベースライン更新アルゴリズム

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。図 71 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。

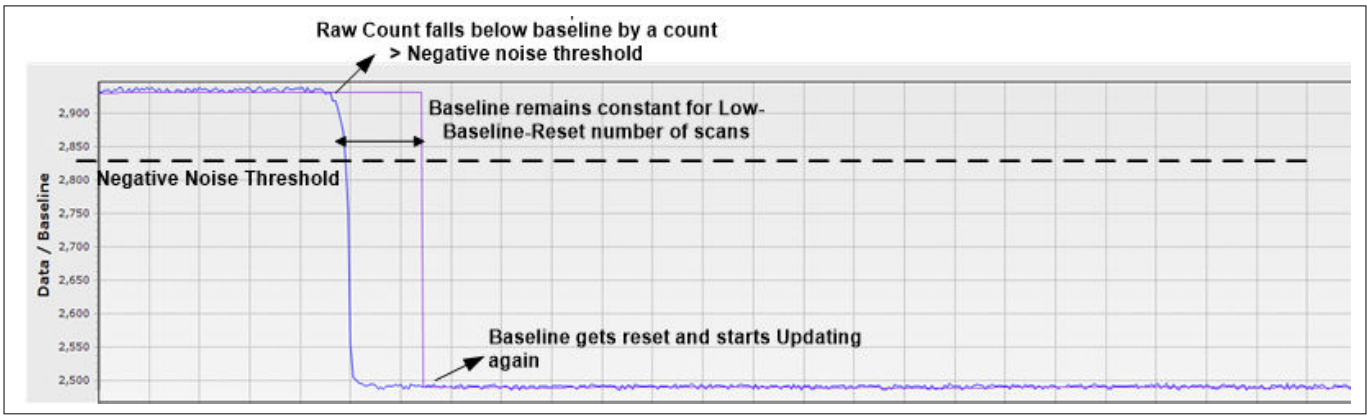


図 71 低ベースライン リセット

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 43 に示すようにセンサーの状態を決定します。

$$\text{Sensor state} = \{ ON \ \ if\ (Signal \geq Finger\ Threshold + Hysteresis) \\ OFF\ if\ \ (Signal \leq Finger\ Threshold - Hysteresis) \}$$

式 43 センサー状態

注: *図 69 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) の差を指します。*

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 パラメーターは指しきい値パラメーターとともに使用され、式 43 と図 72 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 パラメーターは、指しきい値パラメーター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。

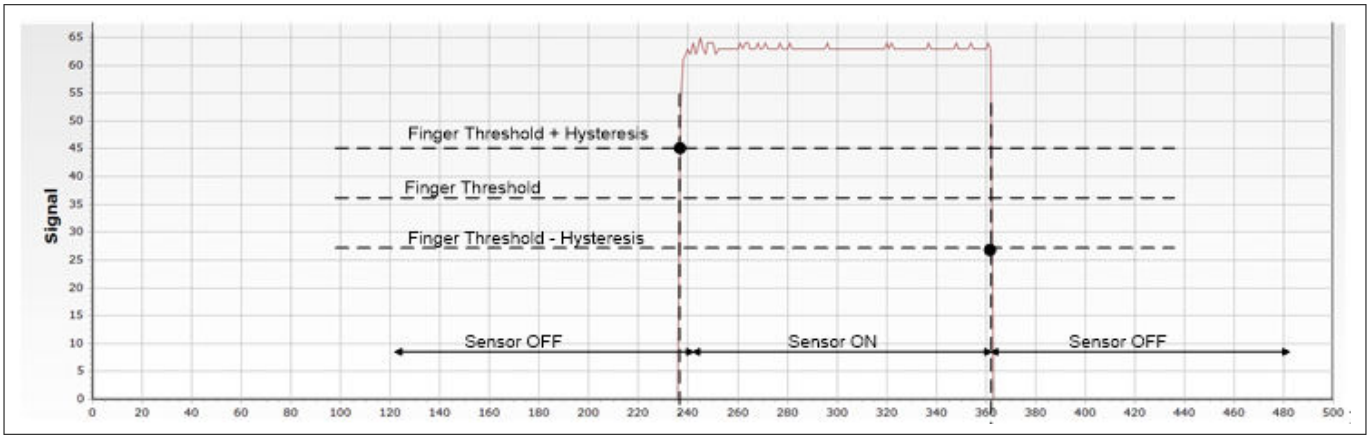


図 72 ヒステリシス

5.3.2.3.5 ノイズしきい値

図 70 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ーは 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(*baseline + noise threshold*) 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 (*finger threshold – hysteresis*) 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ンド電圧の変化があるからです。

一方、ノイズしきい値は (*finger threshold – hysteresis*) 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。これは信号の減少につながり

注: *(signal = raw count – baseline)、センサーの状態がオンとして報告されない場合があります。*

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ーは raw カウントの限界を設定し、図 71 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 パラメーターで指定したサンプル数で更新されません。

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、図 71 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。

注: この場合には、指はセンサーから離れるとき、式 44 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ません。

$$\text{Low Baseline Reset Time} = \frac{Low\ Baseline\ \text{Re}set\ parameter\ value}{Scan\ rate}$$

式 44 低ベースライン リセット時間

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。

$$\begin{aligned}&\text{Sensor state} = \{ON\ if\ (Signal\ \geq\ \text{Finger}\ Threshold\ + Hysteresis)\ for\ scans\ \geq debounce\\&OFF\ if\ (Signal\ \leq\ Finger\ Threshold\ - Hysteresis)\\&OFF\ if\ (Signal\ \geq\ Finger\ Threshold\ +\ Hysteresis)\ for\ scans\ < debounce\}\end{aligned}$$

式 45 デバウンス付きのセンサー状態

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 46 で計算されます。

$$\text{Sensor response time} = \frac{Debounce}{Scan\ Rate}$$

式 46 デバウンスとセンサー応答時間の関係

デバウンス パラメーター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 パラメーター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、ベースラインは図 70 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 (*Baseline – Negative Noise Threshold*) と (*Baseline + Noise Threshold*) 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、図 73 に示すように現在の raw カウントが (*Baseline – Negative Noise Threshold*) よりも高い場合にベースラインは常に更新されます。

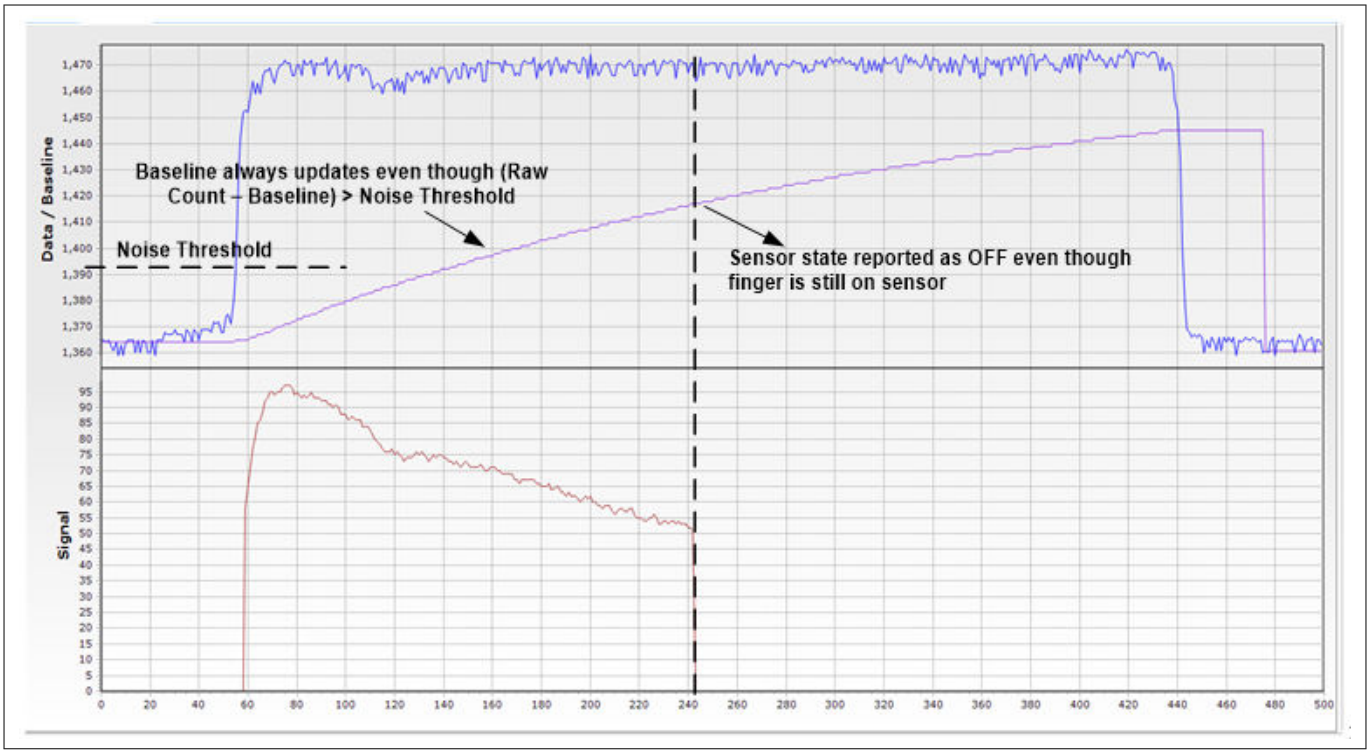


図 73　　センサー自動リセットが有効な場合のベースライン更新

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。このパラメーター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 Full Autotune 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

図 74 に、CSD ボタンの調整手順の概要を示します。

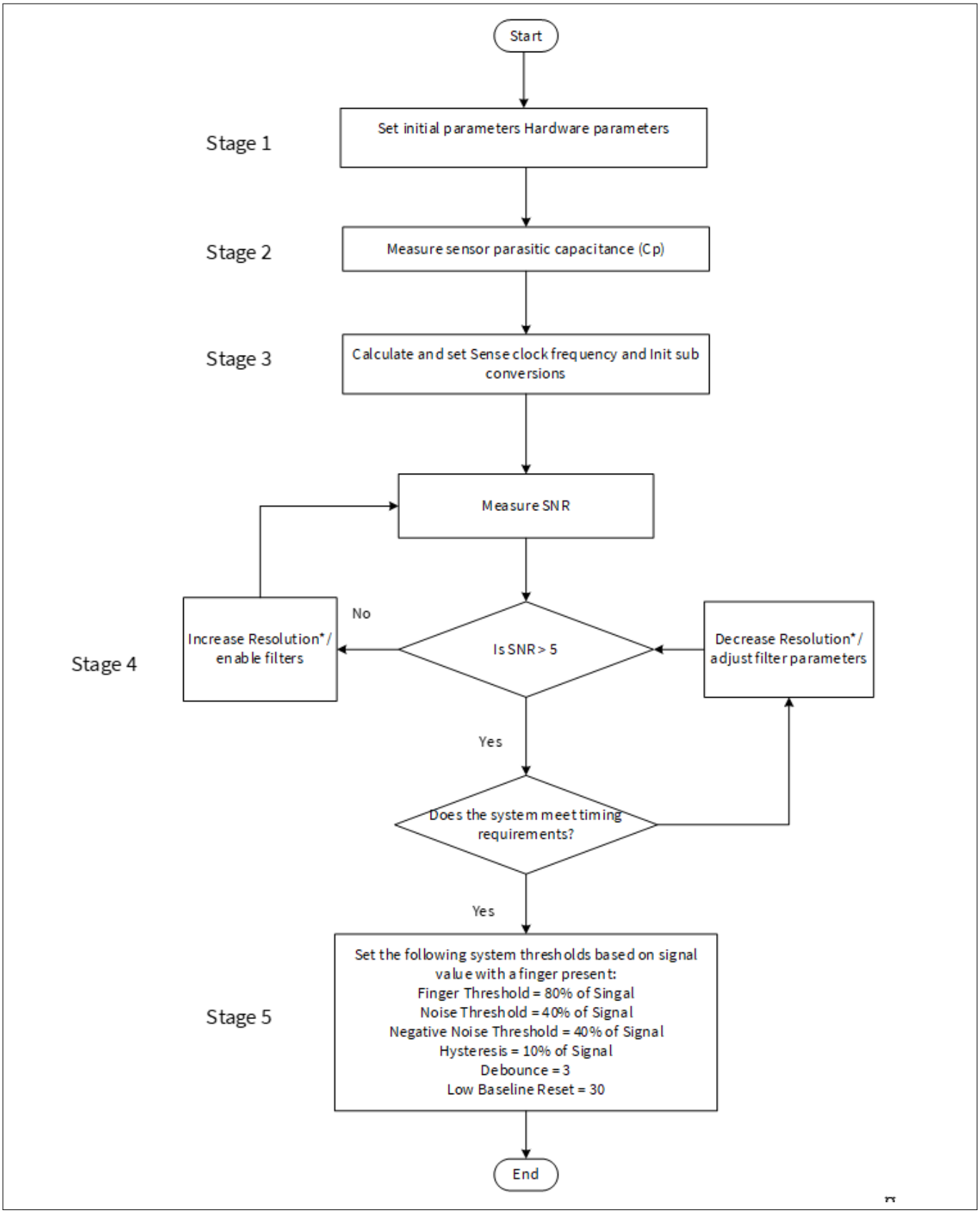


図 74　　CSD ボタン ウィジェット チューニングのフロー チャート

重要:　　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。

セクション第 5.1 章で説明したように、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。図 75 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、図 76 に示すように、重心は非線形になります。

注: *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 の重心が報告されます。*

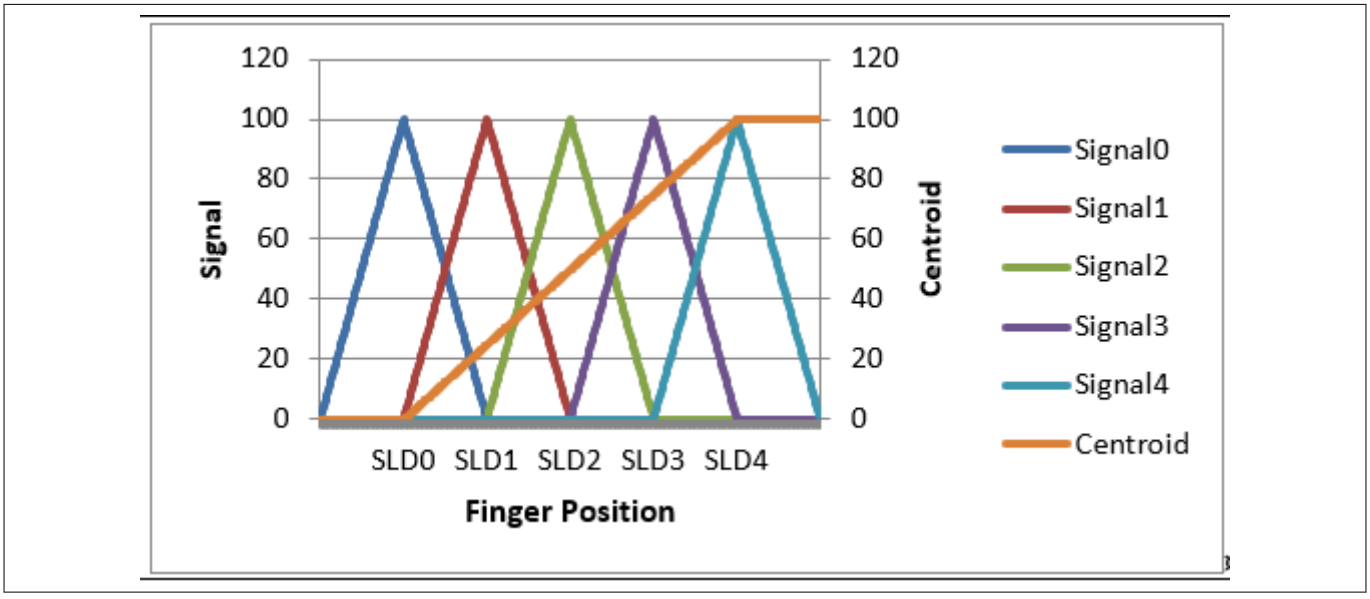


図 75 すべてのスライダー要素の信号が同じ場合の指位置に対する重心の対応

注: *信号 = raw カウント – ベースライン*

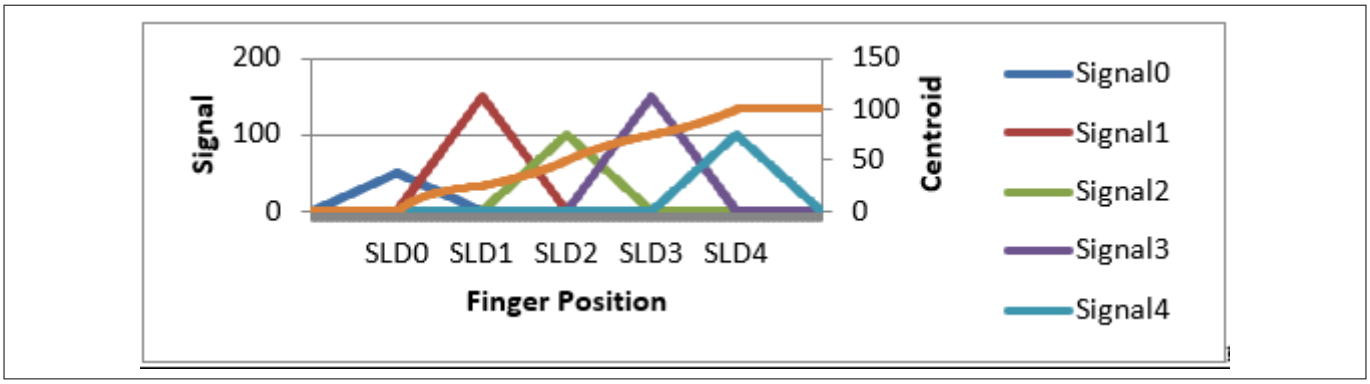


図 76　　すべてのスライダー要素の信号が異なる場合の指位置に対する重心の応答

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 [9]。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 47 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。

$$\text{centroid position} = \left(\frac{S_{x+1} - S_{x-1}}{S_{x+1} + S_{x-1}} + x\right) \times \frac{Resolution}{(n-1)}$$

式 47　　PSOC™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

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差分カウント (ノイズ閾値を除く)

図 77 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。

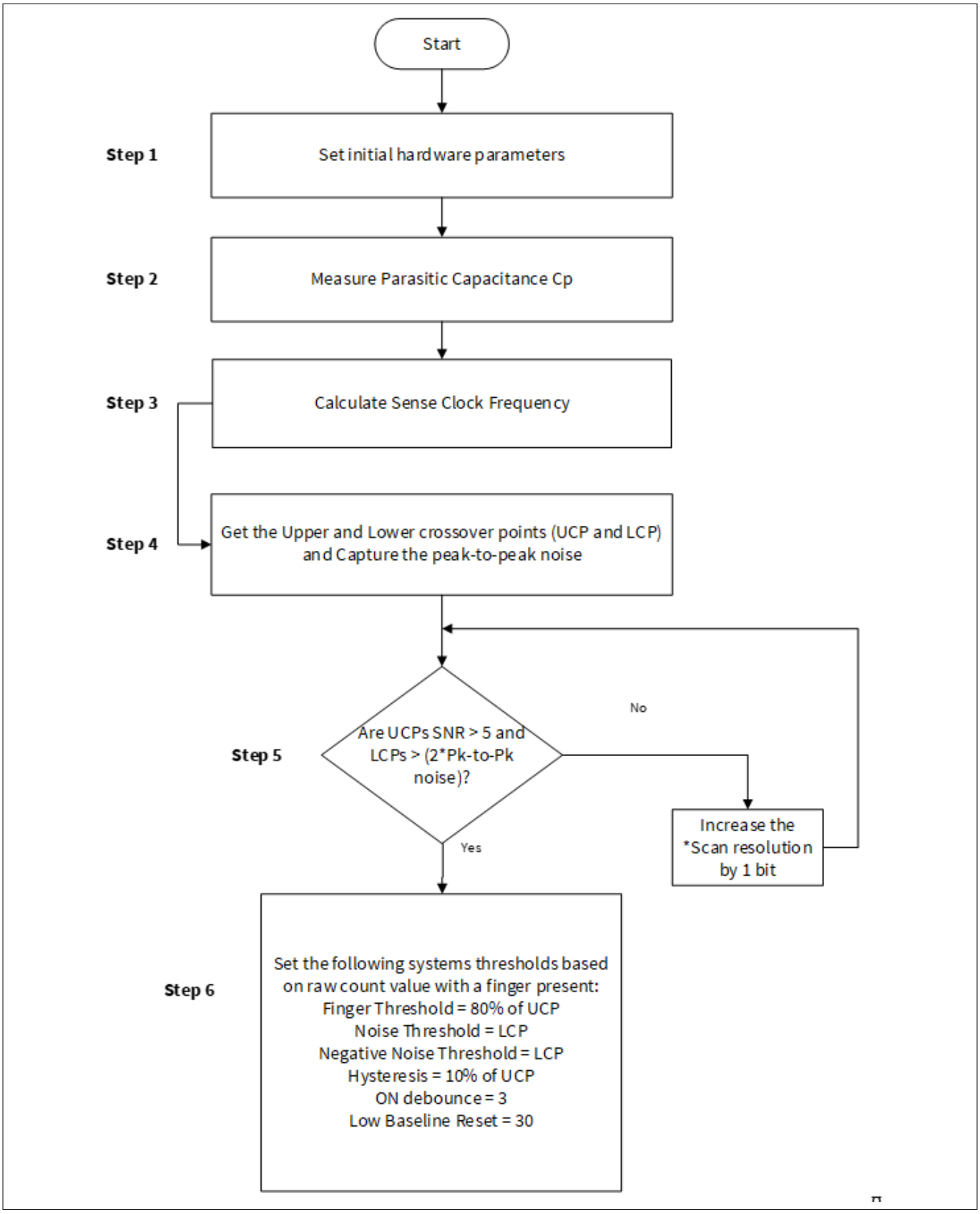


図 77　　CSD スライダー ウィジェット チューニングのフロー チャート

重要:　　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。

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、図 78 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。

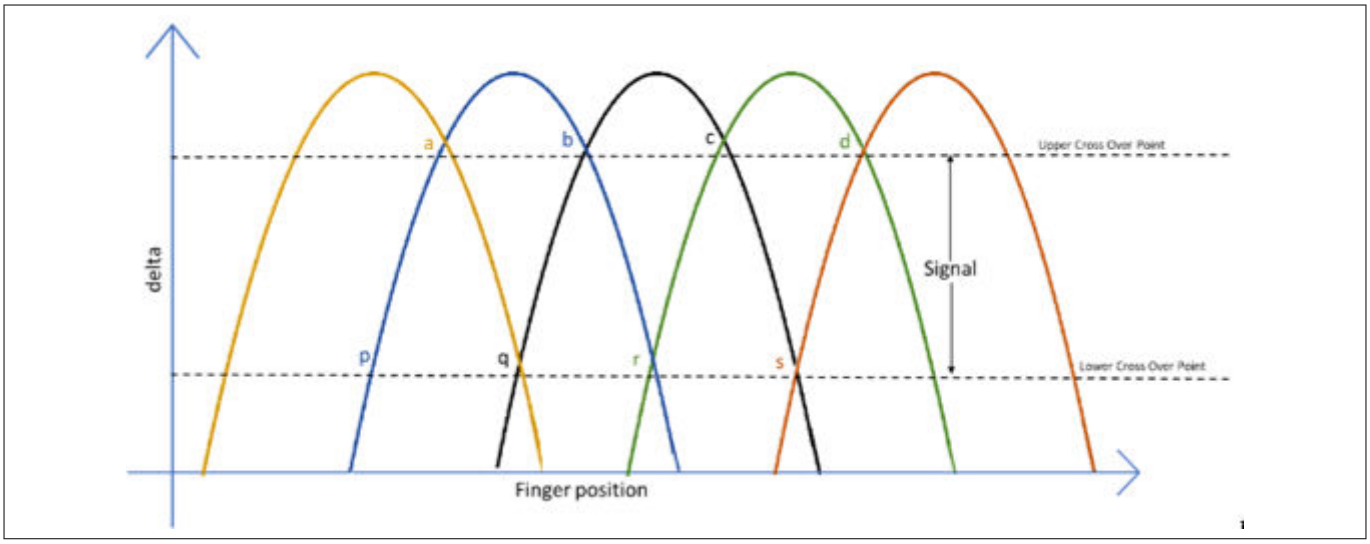


図 78　　差分カウント (デルタ) と指の位置

5.3.2.6　タッチパッドウィジェットの調整

タッチパッドのチューニングガイドラインについては、AN234185 PSOC™ 4 CAPSENSE™ Touchpad design guide を参照してください。

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 - CAPSENSE™による近接センシングを参照してください。

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

相互容量検知システムでは、式 48 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。

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

式 48　　raw カウントとセンサー静電容量の関係

ここで、

G_{CSX} = CAPSENSE™ CSX の容量からデジタル変換利得

C_M = Tx および Rx 電極間の相互容量

図 80 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 49 の変換利得の調整可能なパラメーターは、F_{TX}, N_{MOD}, F_{MOD}, および IDAC です。
この変換利得の概算値は次のとおりです。

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

式 49　　静電容量-デジタル変換利得

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

式 50　　最大カウント式

ここで、
V_{TX} は、図 79 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。

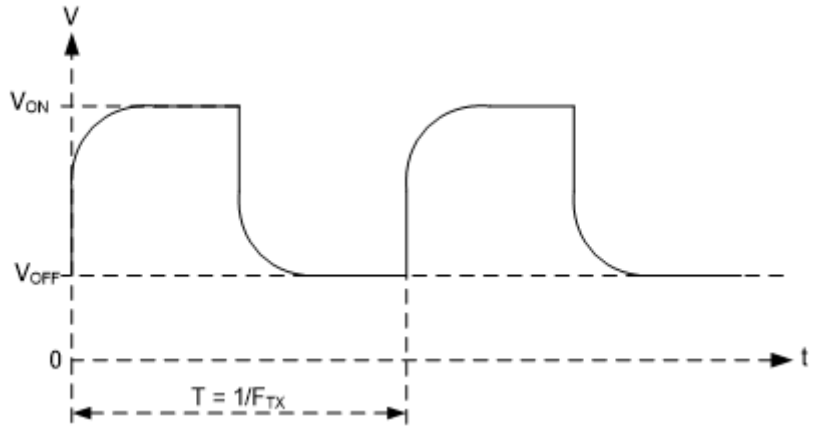


図 79　　CSX センサーの Tx ノードでの電圧

注: *コンポーネントから観測された raw カウントは式 51 で与えられます。$Rawcount_{component}$ 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。*

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

式 51　　$Rawcount_{Component}$

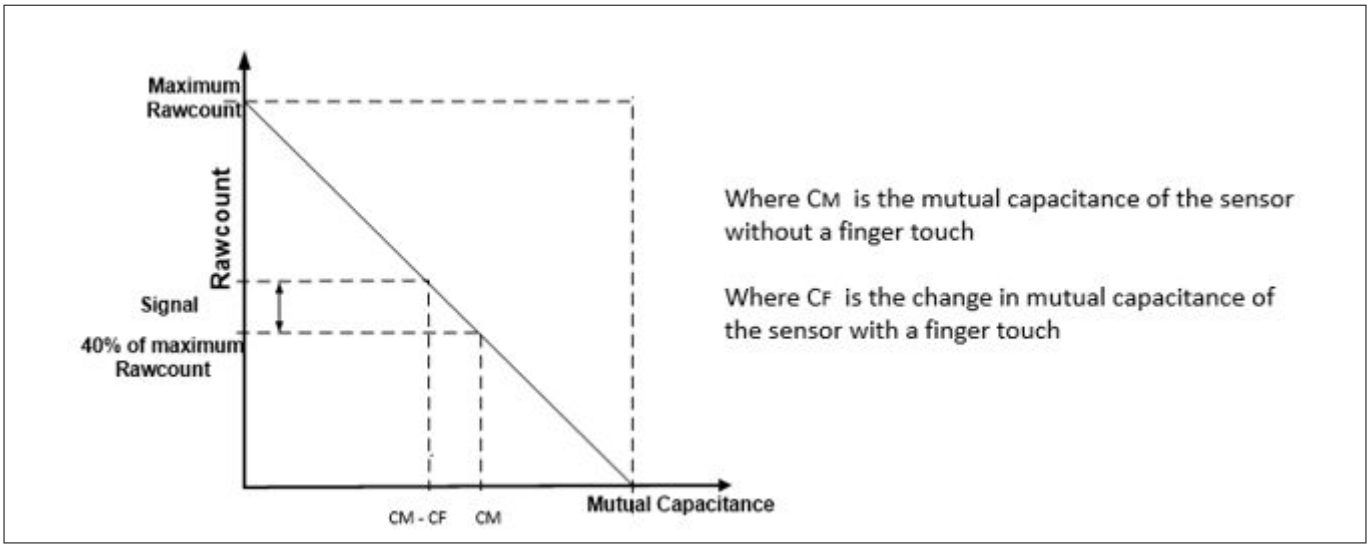


図 80　　raw カウントとセンサー相互静電容量

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。表 15 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。表 15 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 クロックと変調器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 ドキュメントを参照してください。

表 15　　CAPSENSE™信号コンポーネント ハードウェア パラメーター

Sl #	PSOC™ Creator の CAPSENSE™パラメーター	ModusToolbox™での CAPSENSE™パラメーター
1	変調器クロック周波数	変調器クロック分周器
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。

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 52 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 52 を満たすようにを調整します。

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

式 52　　Tx クロック周波数を選択するための条件

式 53 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。

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

式 53　　CSX センサーのスキャン時間

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 ドキュメント/ 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。

FTX 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 FTX 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　変調器クロック周波数

式 40、式 41 に示すように、変調器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。

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。低 C_M の影響を軽減するための設計については、「CSX ボタンの CM が低いことを確認」を参照してください。

5.3.3.2.4 サブ変換回数

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、サブ変換数を増やすと、式 54 に従ってセンサーのスキャン時間も長くなります。

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

式 54 CSX スキャン時間

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段階的に N_{Sub} を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 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

図 81 に、CSX ボタンの調整手順の概要を示します。

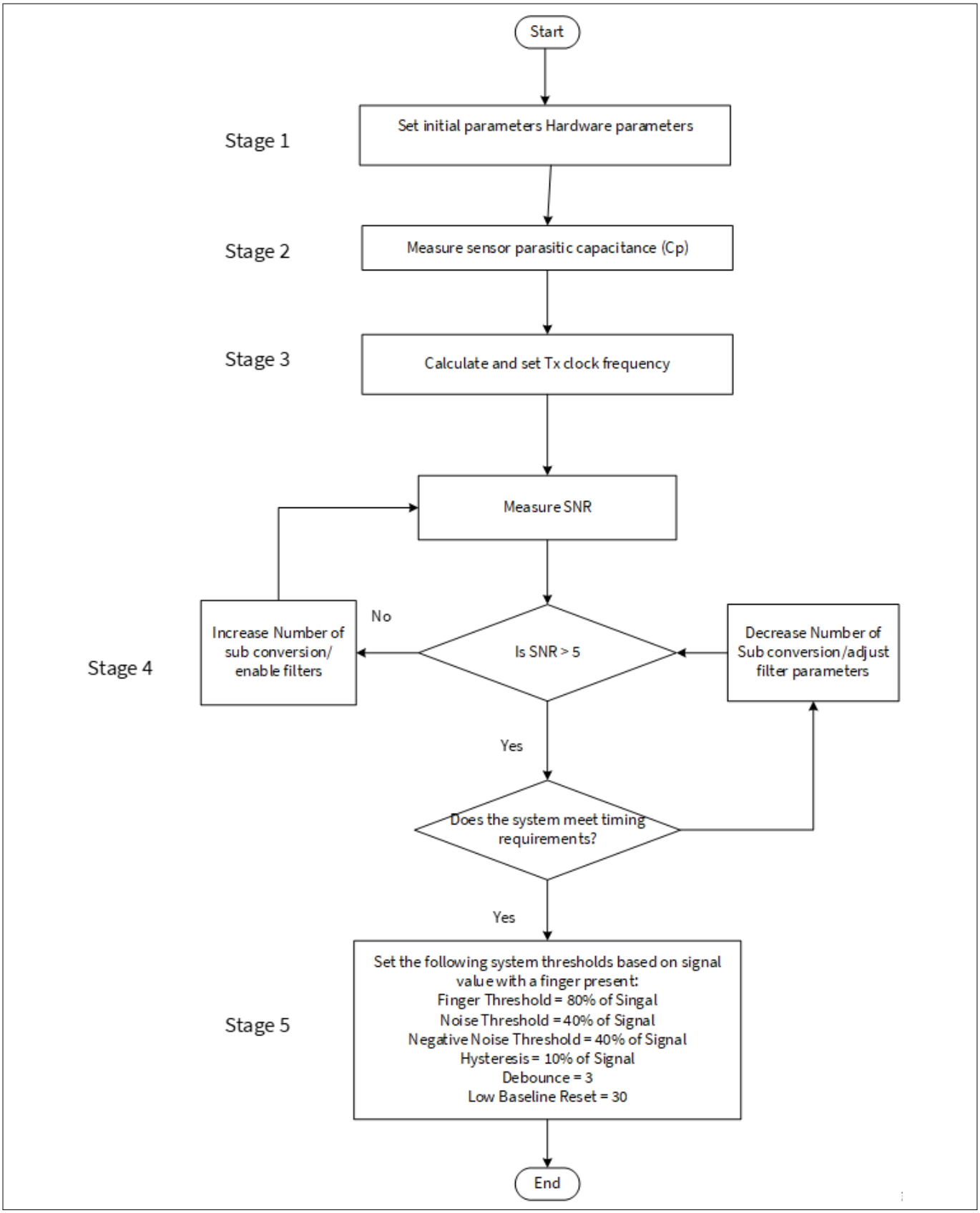


図 81　　CSX ボタン ウィジェットチューニング例

* 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

タッチパッドのチューニングガイドラインについては、AN234185 PSOC™ 4 CAPSENSE™ Touchpad design guide を参照してください。

5.3.4 CSD-RM センシング方式 (第 5 世代および第 5 世代低消費電力)

ここでは、第 5 世代および第 5 世代低消費電力デバイスの CSD-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、CSD-RM センシング方法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。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。

注: 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ります。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、表 10 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。

$$\text{raw count} = G_{CSD}\, C_S$$

式 55 raw カウントとセンサー静電容量の関係

ここで、

C_S = センサーの静電容量

C_S = センサー上に指がないときの C_P

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 55 によるこの変換利得の概算値は、式 56 を使用して示されます。

$$G_{CSD} = MaxCount \frac{1}{SnsClk_{Div} \cdot C_{ref}}$$

式 56　　静電容量-デジタル変換利得

ここで、

式 56 および式 55 による、シングル CDAC モードでの raw カウントの式を式 57 に示します。

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

式 57　　シングル CDAC モードでの raw カウント

ここで、

N_{Sub} = サブ変換回数

$SnsClk_{Div}$ = センス クロック分周器

C_S = センサーの静電容量

C_{ref} = 基準静電容量

$C_{ref} = \text{Ref}CDACCode \times C_{lsb}$

RefCDACCode = 基準 CDAC 値

$C_{lsb} = 8.86fF$

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

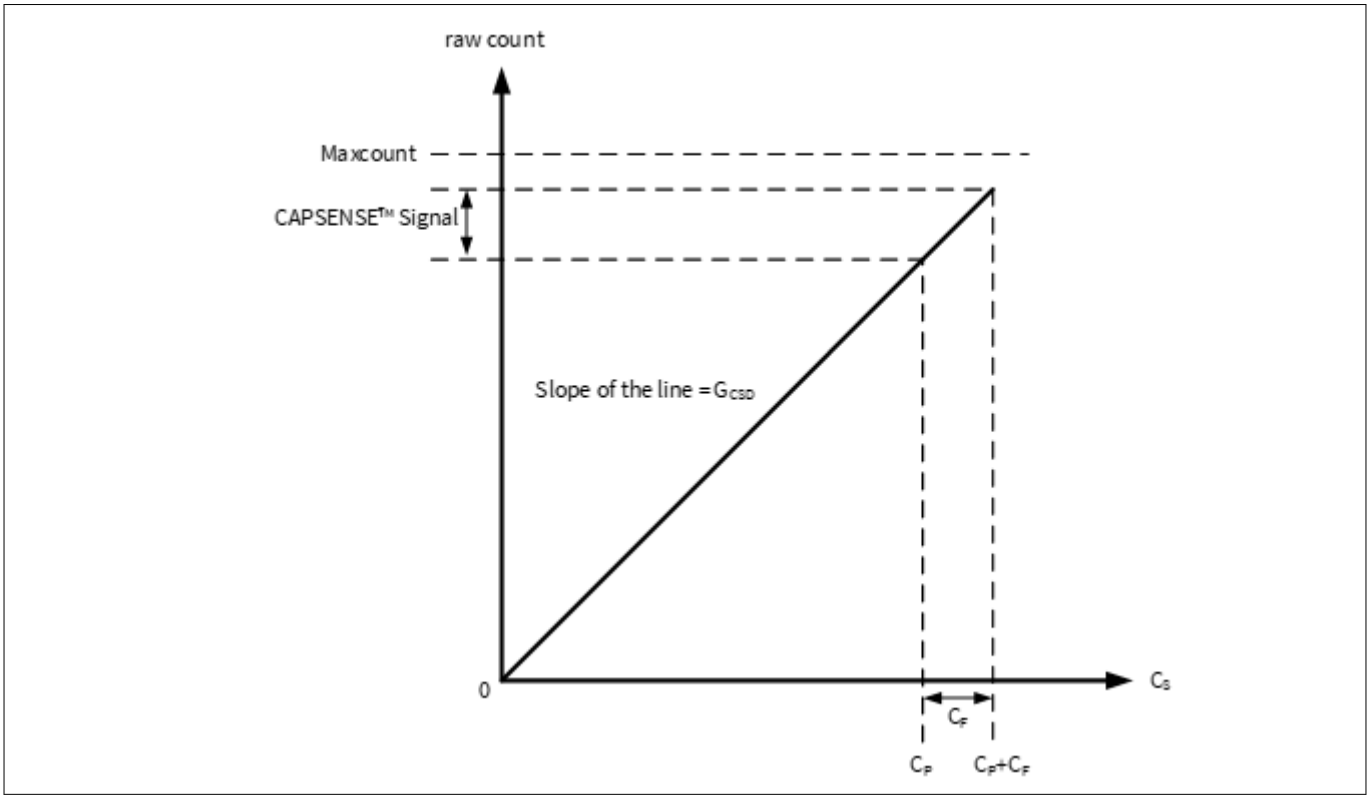


図 82　　センサー容量に対する raw カウント

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。図 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。

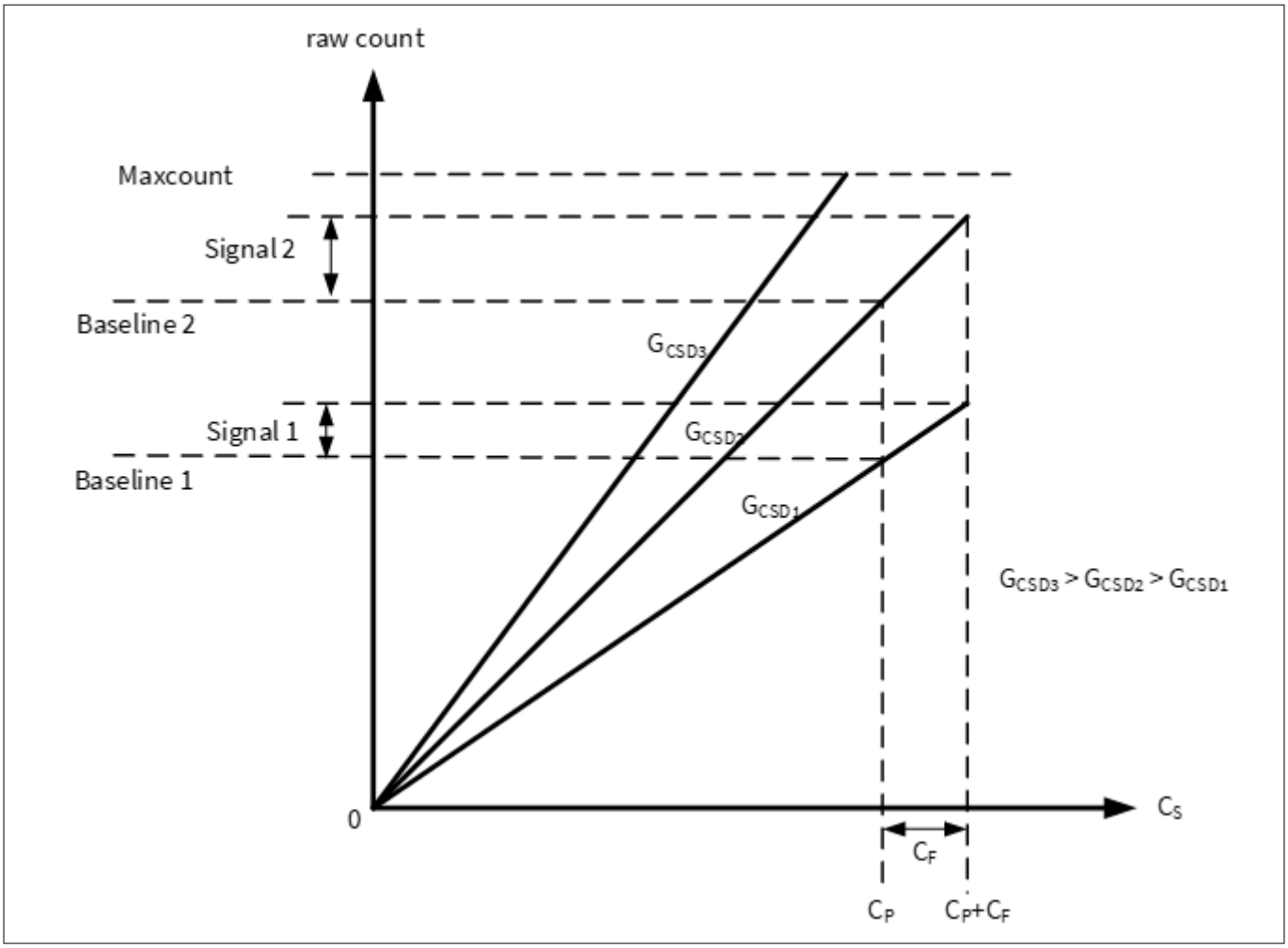


図 83　　さまざまな変換利得の信号値

図 83 に、3 つの変換利得値 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。図 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。

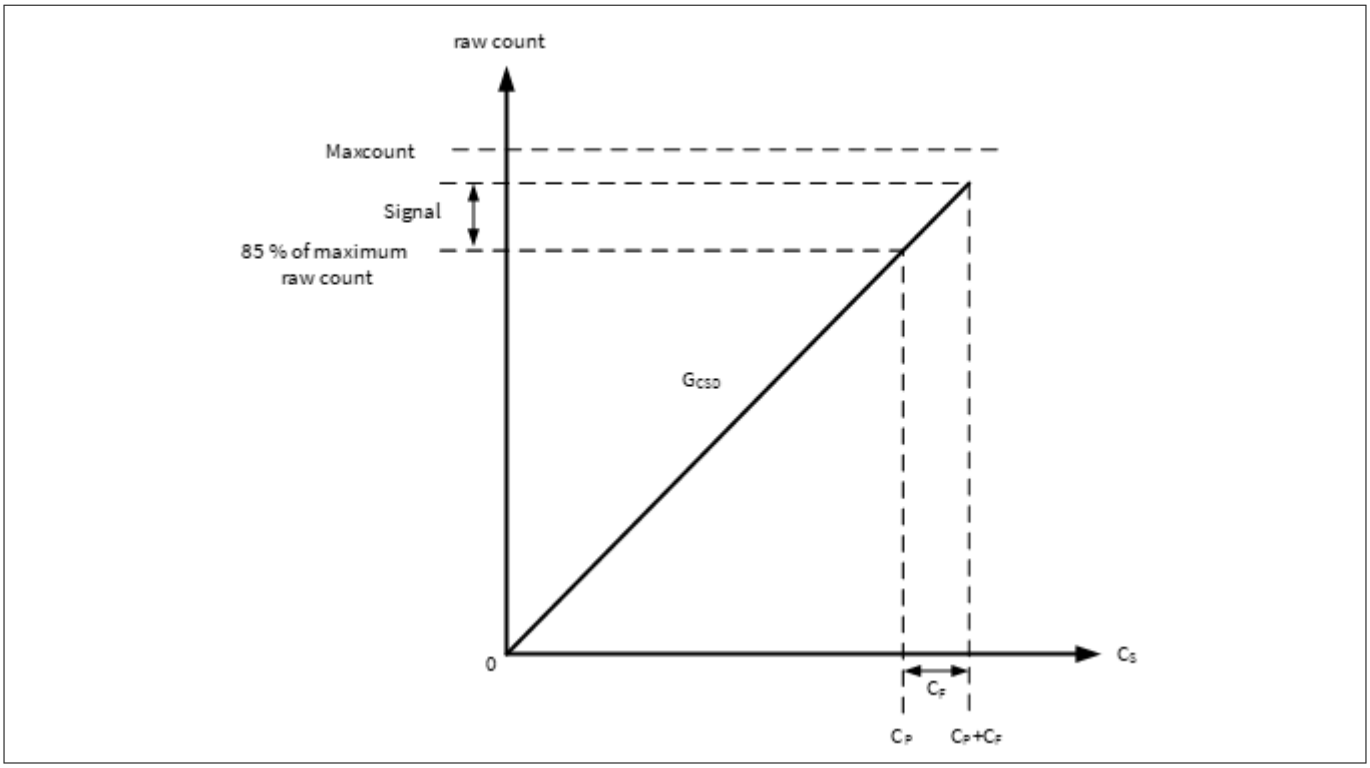


図 84　　推奨チューニング

デュアル CDAC モードでの変換利得

式 19 および式 55 による、デュアル CDAC モードでの raw カウントの式を式 58 に示します。

$$\text{rawcount} = G_{CSD}\,C_S - Maxcount \times \frac{2 \times C_{comp}}{C_{ref} CompCLK_{div}}$$

式 58　　デュアル CDAC モードでの raw カウント

ここで、

MaxCount =N_{Sub} *$SnsClk_{Div}$

$SnsClk_{Div}$ = センス クロック分周器

N_{Sub} = サブ変換回数

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 = センサーの静電容量

C_{ref} = RefCDACCode * C_{lsb}

C_{comp}=CompCDACCode * C_{lsb}

RefCDACCode = 基準 CDAC 値

CompCDACCode = 補償 CDAC 値

C_{lsb} = 8.86fF

G_{CSD} は式 56 で与えられます。

5 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、図 85 に示すように、raw カウント= Maxcount 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。

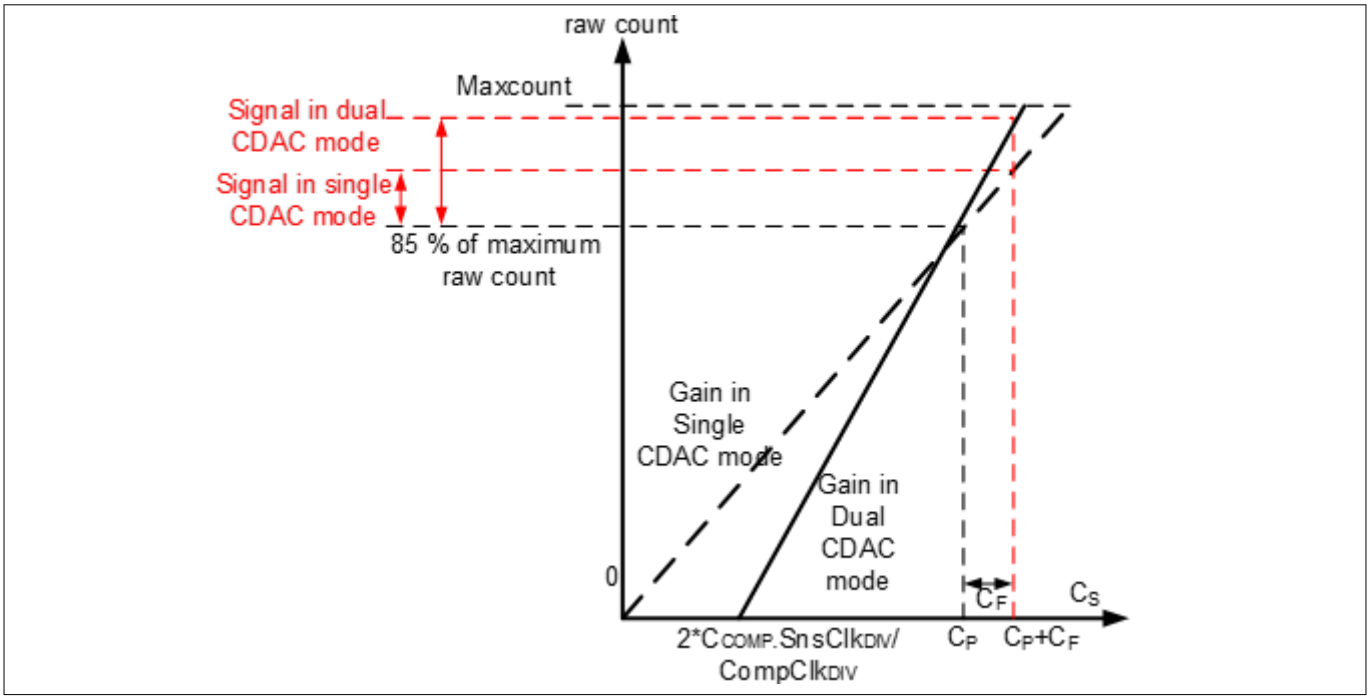


図 85　　デュアル CDAC モードでの推奨されるチューニング

図 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 ($CompClk_{DIV}$ 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 スイッチング ($SnsClk_{DIV}$) 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。

- 補償 CDAC を使用すると、式 59 に示すように、C_S 軸にゼロ以外の切片が導入されます。

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

式 59　　C_{comp} に関する C_S 軸切片

- 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 35 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 35, 式 37, 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。しかし、利得 G_{CSD3} のプロットで図 63 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。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 41 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。
3. 補償の有効化 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 モードで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。常に補償 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。図 64 および図 65 に示すように、CDAC を調整して最高利得を達成しますが、C_P に対応する raw カウントに、温度変化などの環境の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、図 62 と図 65 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 コンバーターは図 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ます。

式 60 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。

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

式 60 フラットスポット幅

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% = raw カウントキャリブレーションパーセンテージ

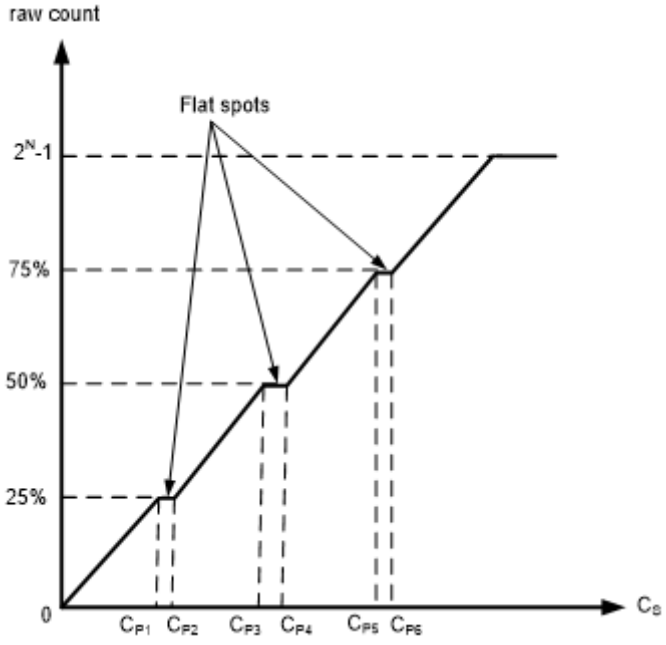


図 86 直接クロック使用時の raw カウント対センサー静電容量のフラット スポット

フラットスポット削減技術

1. raw カウントキャリブレーションを 85%に設定します。

   式 60 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. ディザリングを有効

   追加のディザ 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

   これらのフラットスポットは、直接クロックがセンス 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加

   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。

   表 16 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。

表 16　　フラットスポット削減技術

S. 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%に設定します。	High	感度を向上させる	-
2	ディザリングを有効	High	-	-
3	PRS クロックを有効	Low	EMI/EMC 放射を向上させる	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	-	スキャン時間および消費電力が増加

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。表 17 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™ 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。

表 17 CAPSENSE™コンポーネント ハードウェア パラメーター

S. No	ModusToolbox™での CAPSENSE™パラメーター
1	スキャンモード
2	スキャン接続方式
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	Fine CDAC 値
11	サブ変換回数
12	CDAC ディザを有効

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™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 ドキュメント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

表 17 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してください。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。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。

調整手順は SmartSense セクションを参照してください。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てください。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。

図 87 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。

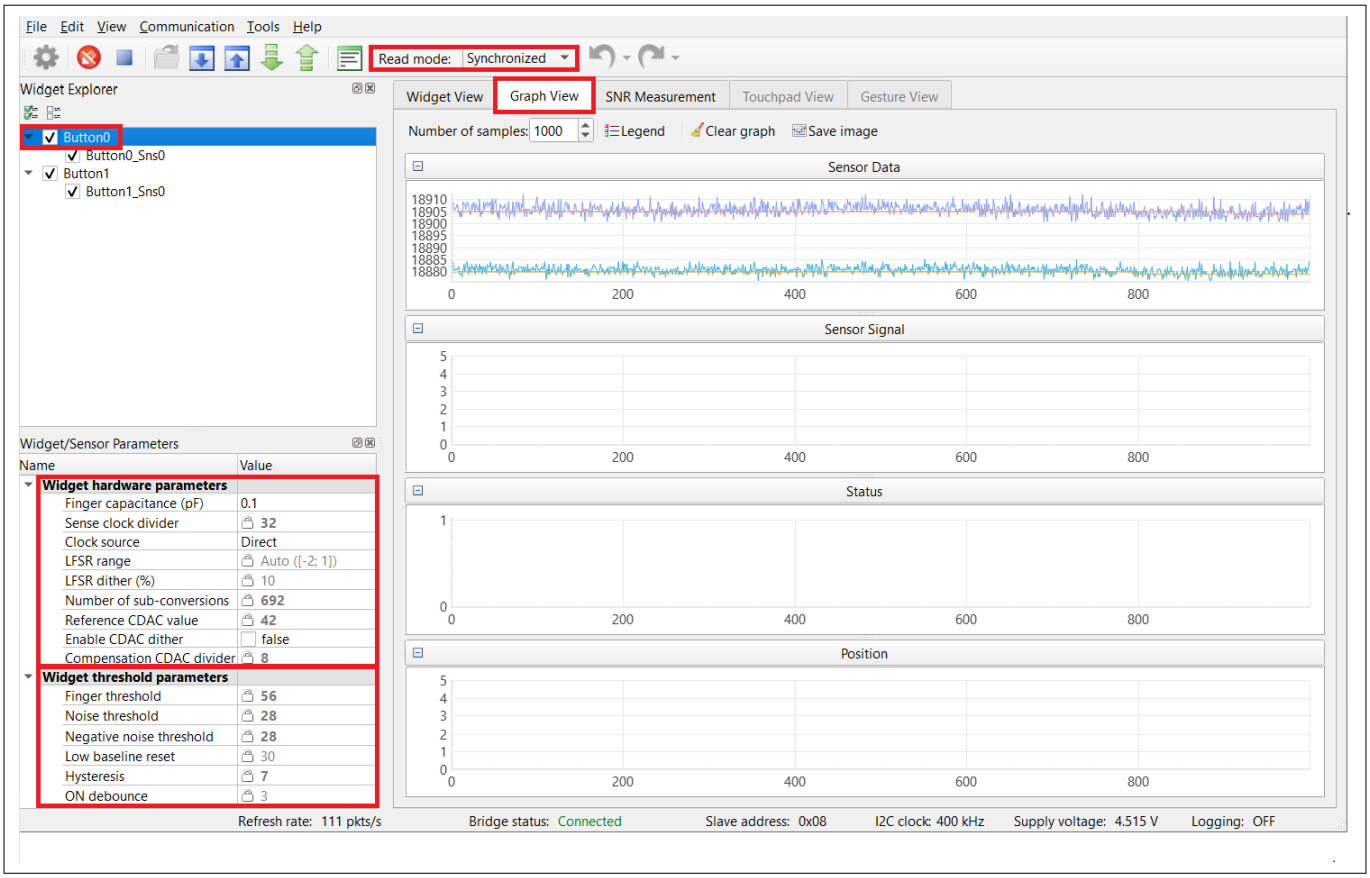


図 87　　チューナー GUI でのハードウェアパラメーター値の読み戻し

5.3.4.2.2　ハードウェア パラメーターの手動チューニング

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。

第 5 世代の低消費電力 CAPSENSE™では、自律スキャンが本質的にサポートされており、CPU の介入を必要とせず、CTRLMUX と DMA を必要としません。

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。

第 5 世代の CAPSENSE™テクノロジでは、CTRLMUX によって限られたピン数がサポートされていましたが、第 5 世代低消費電力 CAPSENSE™では、すべての CAPSENSE™ピンが AMUXBUS 経由で自律スキャンモードをサポートします。

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
- 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

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てください。

必要な変調器クロック周波数 (F_{MOD}) に基づいて、式 61 を使用して変調器クロック分周器を計算します。

$$\text{Modulator clock divider} = \frac{F_{Clock}}{F_{MOD}}$$

式 61　　変調器クロック分周器

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 (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 raw カウントは破棄されます。

初期化サブ変換数は、式 62 に基づいて選択する必要があります。

$$\text{Number of init subconversions} = ceiling\left(\frac{C_{MOD} \times V_{OS}}{VDDA \times C_S \times \left(1 - Base\,\%\right) \times \left(\frac{1}{Bal\,\%} - 1\right)}\right) + 1$$

または

$$\text{Number of init subconversions} = ceiling\left(\frac{C_{MOD} \times V_{OS}}{VDDA \times SnClkDiv \times C_{ref} \times (1 - Bal\,\%)}\right) + 1$$

式 62　　初期化サブ変換数

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)

***注**:　　車載グレード (E グレード) のデバイスには Vos=4mV を使用してください。*

C_S = センサーの静電容量

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

$SnsClk_{Div}$ = センス クロック分周器

C_{ref} = 基準静電容量

C_{ref} = RefCDACCode * C_{lsb}

RefCDACCode = 基準 CDAC 値

C_{lsb} = 8.86 fF

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

「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てください。

- 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, PSOC™ 4000T, 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直接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。

PRS/SSC はエミッションを低減するために使用できますが、高い Nsubs を必要とするため消費電力が増加します。最大拡散のためには、Nsubs を 1024 (または 1024 の倍数) に設定することを推奨します。Nsubs が 1024 より小さい、または 1024 の倍数でない場合、パフォーマンスは最適化されません。

センス クロック分周器

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。

これは、オシロスコープとアクティブプローブを用いてセンサーの充放電波形を観察することで検証できます。センサーの測定は、センス ピンや直列抵抗ではなく、電極の近くで行ってください。

***注**: 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 回行われます。*

センサー/シールドで観測された波形については、図 88 および図 89 を参照してください。図 88 は、センスクロック周波数が正しく調整されている場合、適切な充電を示します。パルス幅は少なくとも 5Tau です。つまり各フェーズの終了時に電圧は必要電圧の少なくとも 99.3%に達しています。図 89 は、不完全な整定 (充放電) を示します。充電が不完全な場合は、センスクロック分周を増やしてください。

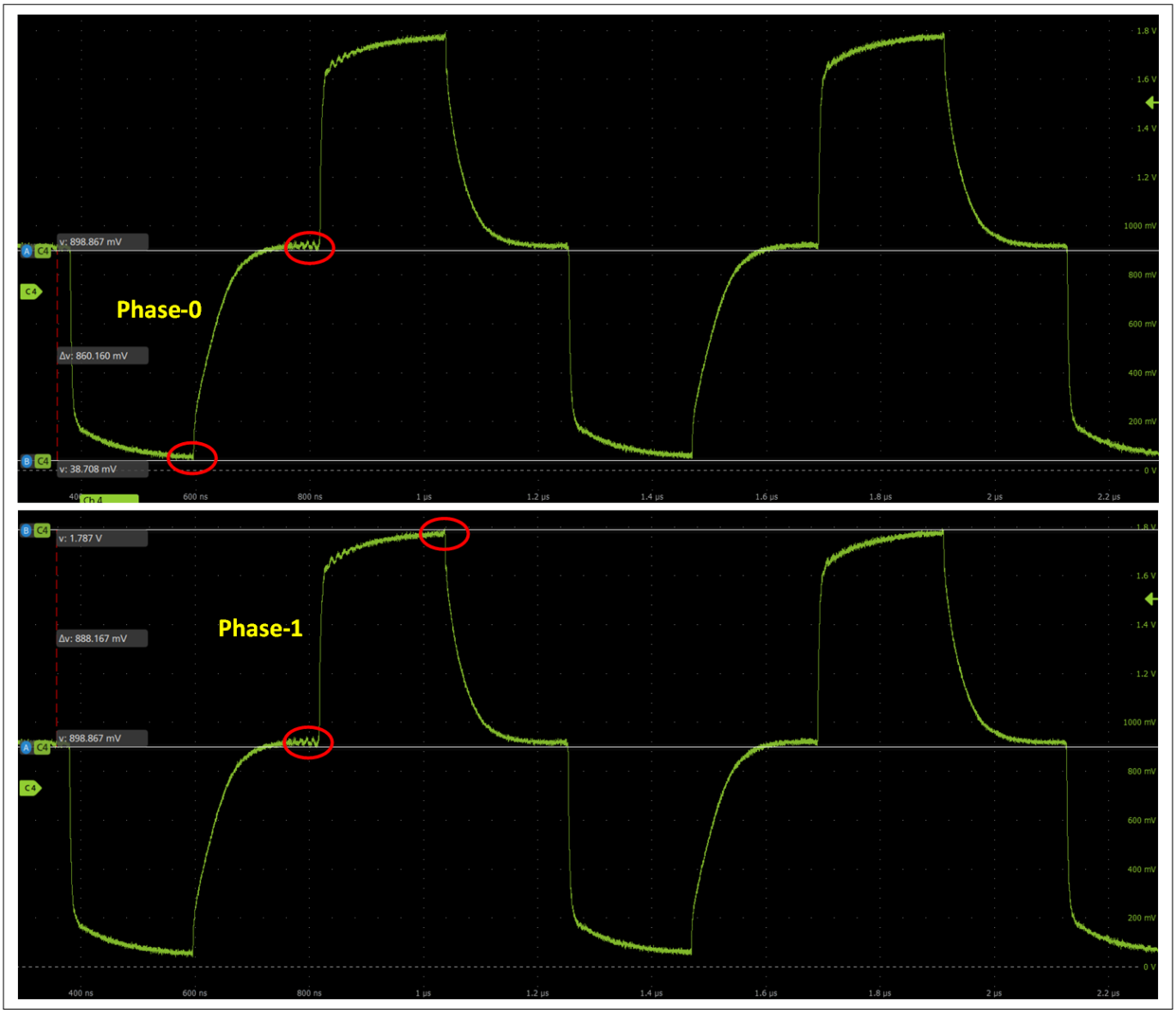


図 88 センサーの適切な充電サイクル

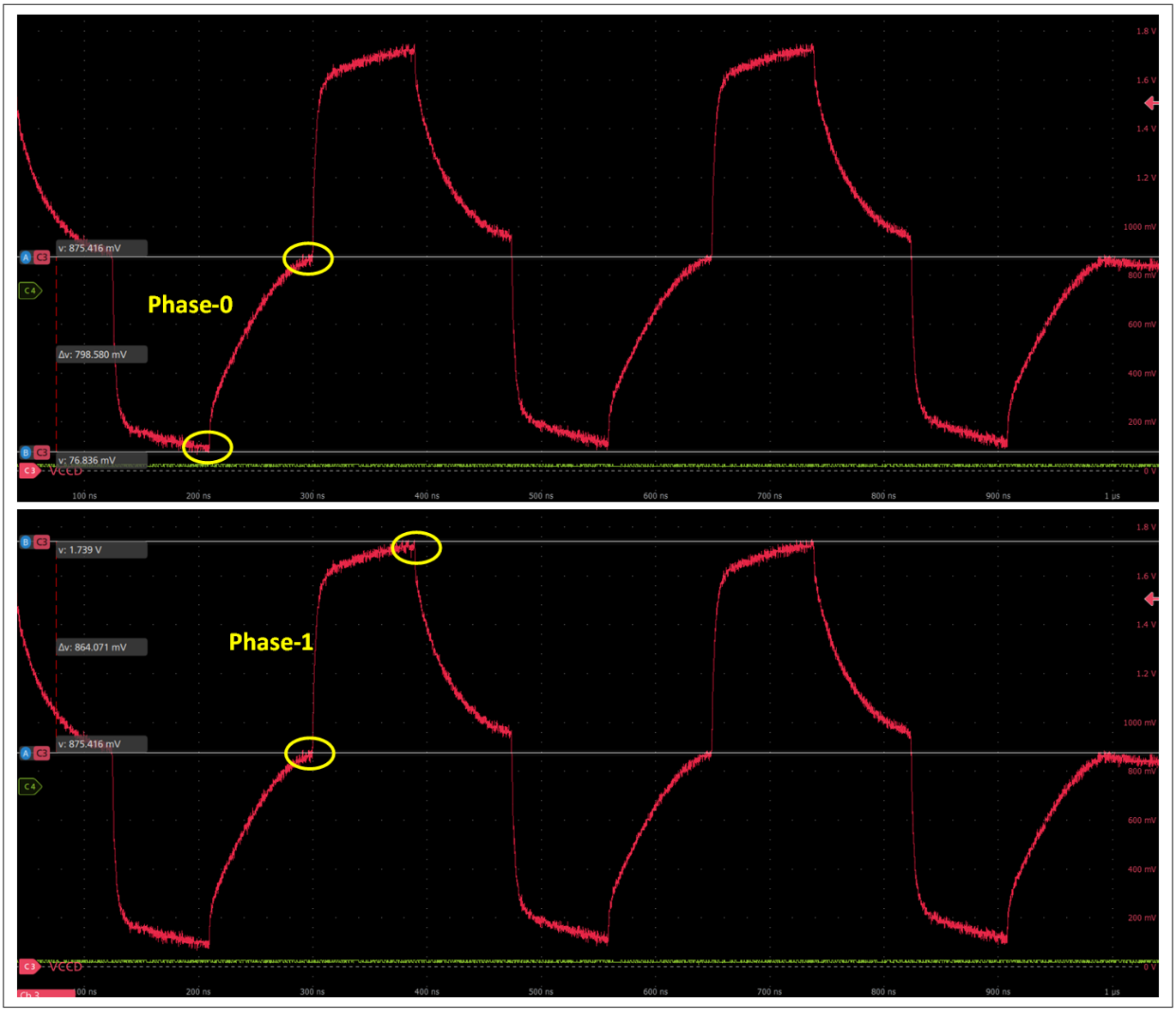


図 89　　センサーの不適切な充電サイクル

*注:　　センスクロック分周器は **4 で割り切れる**必要があります。これにより、4 つのスキャンフェーズの時間がすべて同じになります。*

全てのセンサーとシールドに対して同じ手順を繰り返してください。各センサが完全に充放電するためには、異なるセンスクロック分周値が必要な場合があります。しかし、同じスキャンスロットにあるすべてのセンサーは、同じセンスクロックソース、センスクロック分周器、サブコンバージョン数を持つ必要があります。したがって、与えられたスキャンスロットで最大のセンスクロック分周を取り、そのスロットを共有する他のすべてのセンサーに適用してください。センスクロック分周器が高くなると、充電と放電サイクルが増えるため、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。

$$\text{Sense clock divider} = \frac{F_{MOD}}{F_{SW}}$$

式 63　　センス クロック分周器

式 56 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、図 48 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。

サブ変換回数

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 64 に従ってセンサーのスキャン時間も長くなります。

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

式 64　　CSD-RM スキャン時間

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 Nsub 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を増やしてみてください。

注:　CDAC ディザリングが最適に機能するように、サブコンバージョン数は 8 以上でなければいけません。PRS/SSC を使用する場合は、最適なパフォーマンスを得るために 1024 の倍数にする必要があります。

容量性 DAC

第 5 世代 CAPSENSE™は 3 つの CDAC をサポートします。3 つの CDAC である、基準 CDAC（C_{ref}）、Fine CDAC (Cfine)、補償 CDAC（C_{comp}）は、図 44 が示すように、C_{MOD}'のバランスをとります。これらは、静電容量からデジタルへの変換のための変換利得および 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
- 自動キャリブレーションが無効の場合、基準および補償 CDAC の補償 CDAC 分周器と DAC コードを手動で設定可能です。

表 18 は CAPSENSE™コンフィギュレータで設定できる有効な CDAC の組み合わせを示します。

表 18　　有効な CDAC 構成

Sr. no.	C_{ref}	C_{fine}	C_{comp}
1	自動	自動	自動
2	自動	自動	無効
3	オート	無効	自動
4	自動	無効	無効
5	設定値	設定値	自動
6	設定値	設定値	設定値
7	設定値	設定値	無効
8	設定値	無効	自動
9	設定値	無効	設定値
10	設定値	無効	無効

(続く)

表 18 (続き) 有効な CDAC 構成

Sr. no.	C_{ref}	C_{fine}	C_{comp}
11	無効	自動	自動
12	無効	自動	無効
13	無効	設定値	自動
14	無効	設定値	設定値
15	無効	設定値	無効

***注**: CDAC の場合、自動キャリブレーションを常に有効にすることを推奨します。これにより、最適なチューニングを確保できます。*

C_{ref} ブーストパラメーターは、基準 CAPDAC の感度を調整するために使用できます。これは、補償 CDAC (C_{comp}) とリファレンス CAPDAC が「自動」に設定されている 1 と 3 の組み合わせでのみ使用できます。

基準 CDAC (Cref)

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、センサーの相互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 スイッチングが必要です。

C_{ref} は、以下の方程式を使用して計算します。

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

ここで、

RefCDACCode = 基準 CDAC 値

C_{lsb} = 8.86 fF

C_{ref} は以下の基準を満たす必要があります。

$$\text{RefCDACCode} > 6$$

自動校正された RefCDACCode が 6 より小さい場合は、Fine CDAC を有効にします。

RefCDACCode を増やすためには、以下のオプションを使用できます。

- センスクロック分周を下げてください。センスクロックのパルス幅は、センサー容量が完全に充放電するのに十分な長さでなければいけないことに注意してください。このセンスクロックの低下は、直列抵抗を小さくするか、シールド Cp を小さくすることで対応できます。
- 補正を無効にしてください

Fine 基準 CDAC (Cfine)

第 5 世代の低消費電力 CAPSENSE™は、Fine CDAC で構成されます。Fine CDAC はプログラマブル CDAC で、基準 CDAC のより微細な分解能を達成するために使用されます。したがって、Cfine を Cref とともに有効にすることで、精度が向上します。Fine CDAC の場合、自動キャリブレーションを常に有効にすることを推奨します。これにより最適なパフォーマンスが保証されます。

C_{fine} = Fine 基準 CDAC = FineCDACCode * C_{lsb}

FineCDACCode = Fine CDAC 値

C_{lsb} = 2.2 nF

補償 CDAC (Ccomp)

補償 CDAC を有効にすることは「デュアル CDAC」モードと呼ばれ、変換利得および 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、以下の式 65 に基づいて CDAC 補償分周器を選択します。

1. CDAC 補償分周器 > = 4
2. K_{comp} は整数である必要があります。

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

式 65　　CDAC 補償分周器

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

  ***注**: raw カウントが 85% で飽和している場合、飽和を避けるためにキャリブレーションパーセンテージをさらに下げることができます。*

- フラット スポットの影響を減らす

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 55, 式 56, および式 58 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。

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構成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。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。

最適なパフォーマンスを確保するため、常に自動キャリブレーションモードで CDAC ディザスケールを有効にすることを推奨します。CDAC ディザスケールを手動で設定する必要がある場合は、以下の表で一般的な推奨値を確認してください。

表 19　　CSD センサーのディザスケール推奨値

寄生容量 (C_p)	スケール
2 pF <= C_p < 3 pF	3
3 pF <= C_p < 5 pF	2
5 pF <= C_p < 10 pF	1
C_p >= 10 pF	0

ディザ関連パラメータの設定の詳細については、PSOC™ 4: MSCLP CAPSENSE™ low power (section '**Stage 2: Measure sensor capacitance to set CDAC Dither parameter**')を参照してください。

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ー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™ CSDRM シールド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。波形の例を図 90 と図 91 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。

***注**:　　センスクロック分周器は **4 で割り切れる**必要があります。これにより、4 つのスキャンフェーズの時間がすべて同じになります。*

センスクロック分周器が高くなると、充電と放電サイクルが増えるため、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。良好な耐液性を確保するため、最大シールド周波数（F_{Shield}）のおおよその値は、シールド波形の正しい充電と放電を保証します。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。

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。

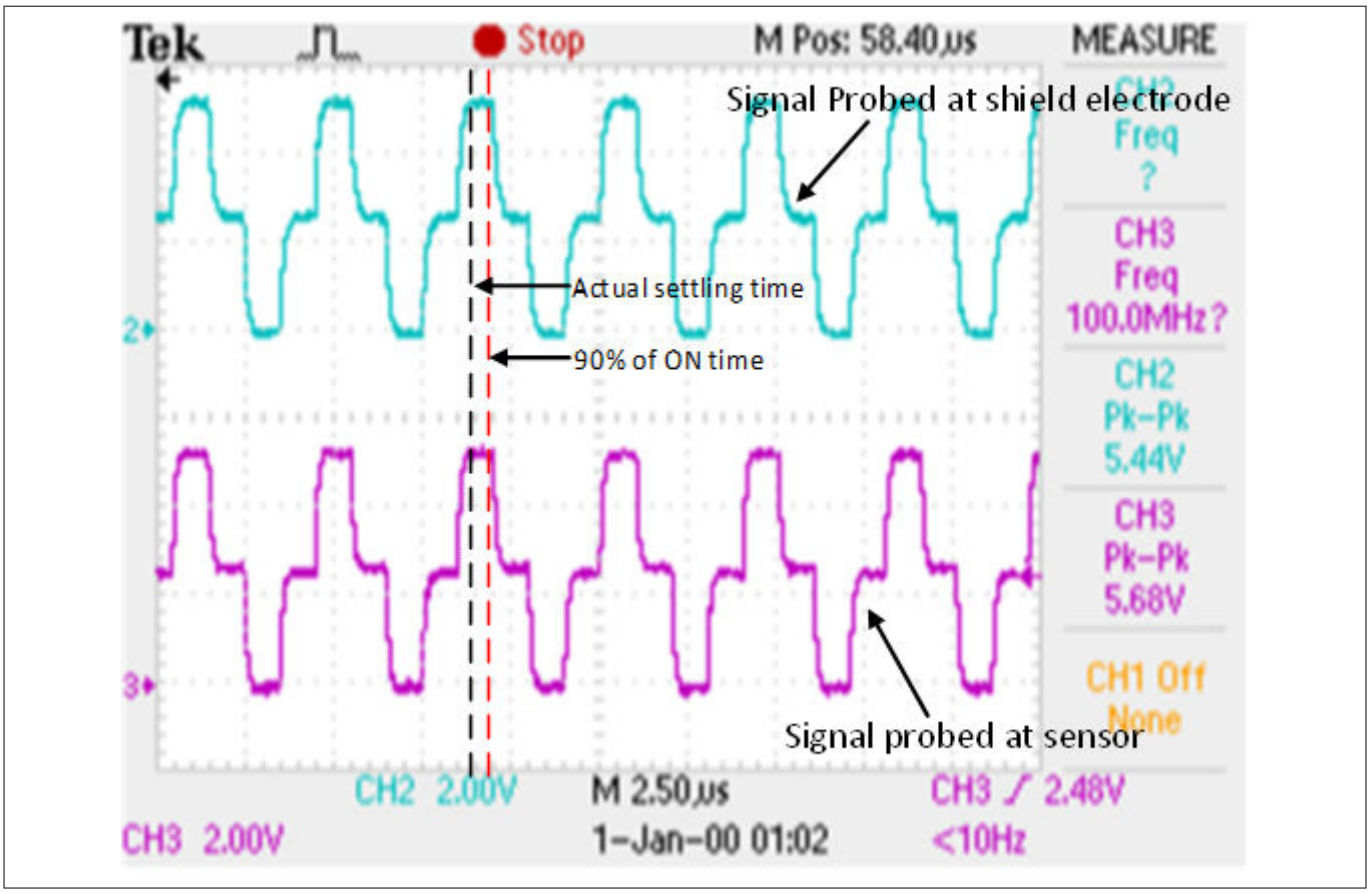


図 90 適切に調整されたシールド波形 (アクティブシールド)

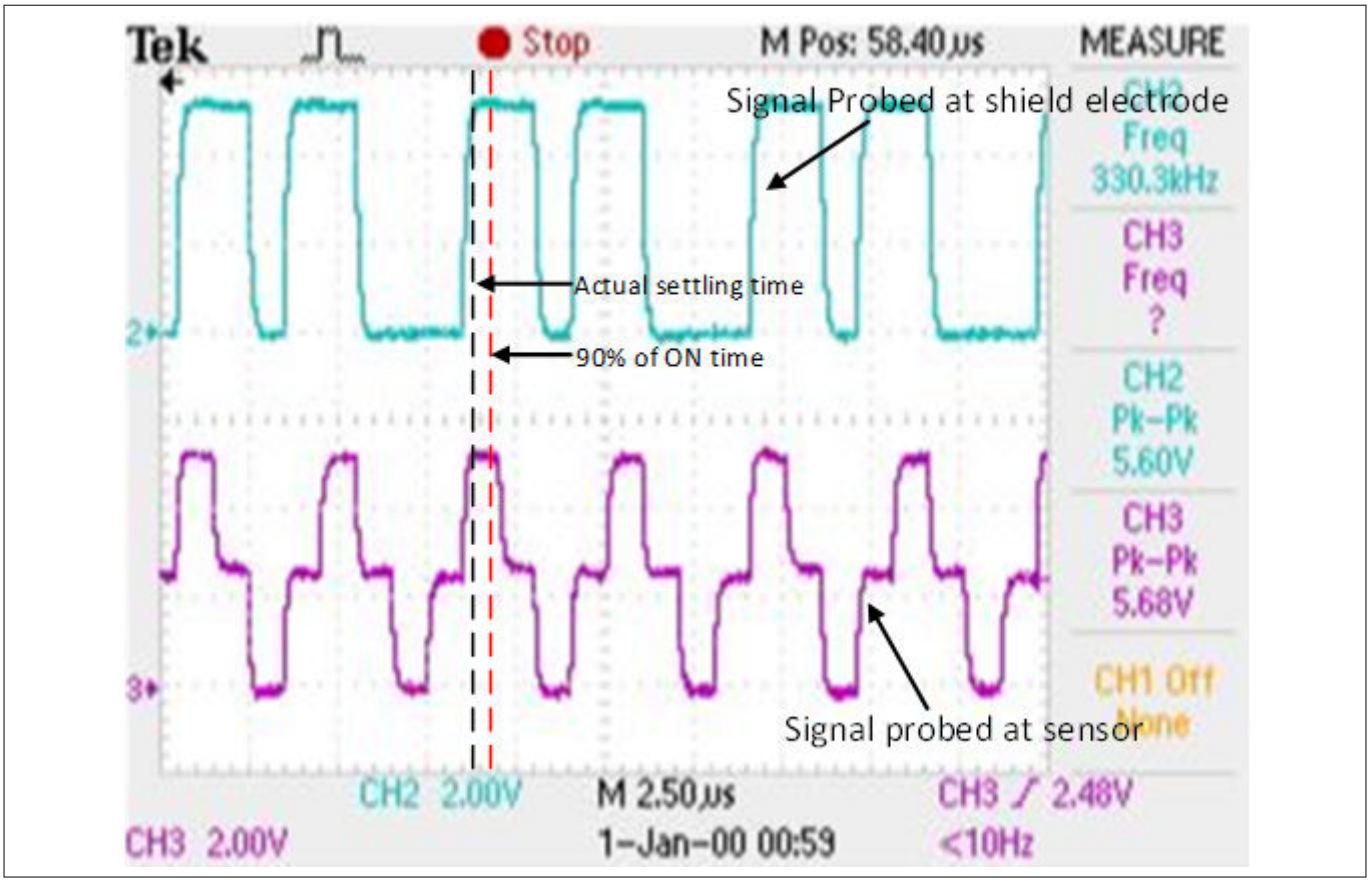


図 91　　適切に調整されたシールド波形 (パッシブシールド)

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。

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™ CSDRM シールドのセクションを参照してください。

5.3.4.2.4 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。

5.3.4.2.5 自律スキャンの設定

自律スキャンは、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。図 92 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、自律スキャンと割込み駆動スキャンの要件を備えた CAPSENSE™ CPU 帯域幅を示します。自律スキャンでは、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 割込みが発生することはありません。ただし、割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 割込みが発生します。

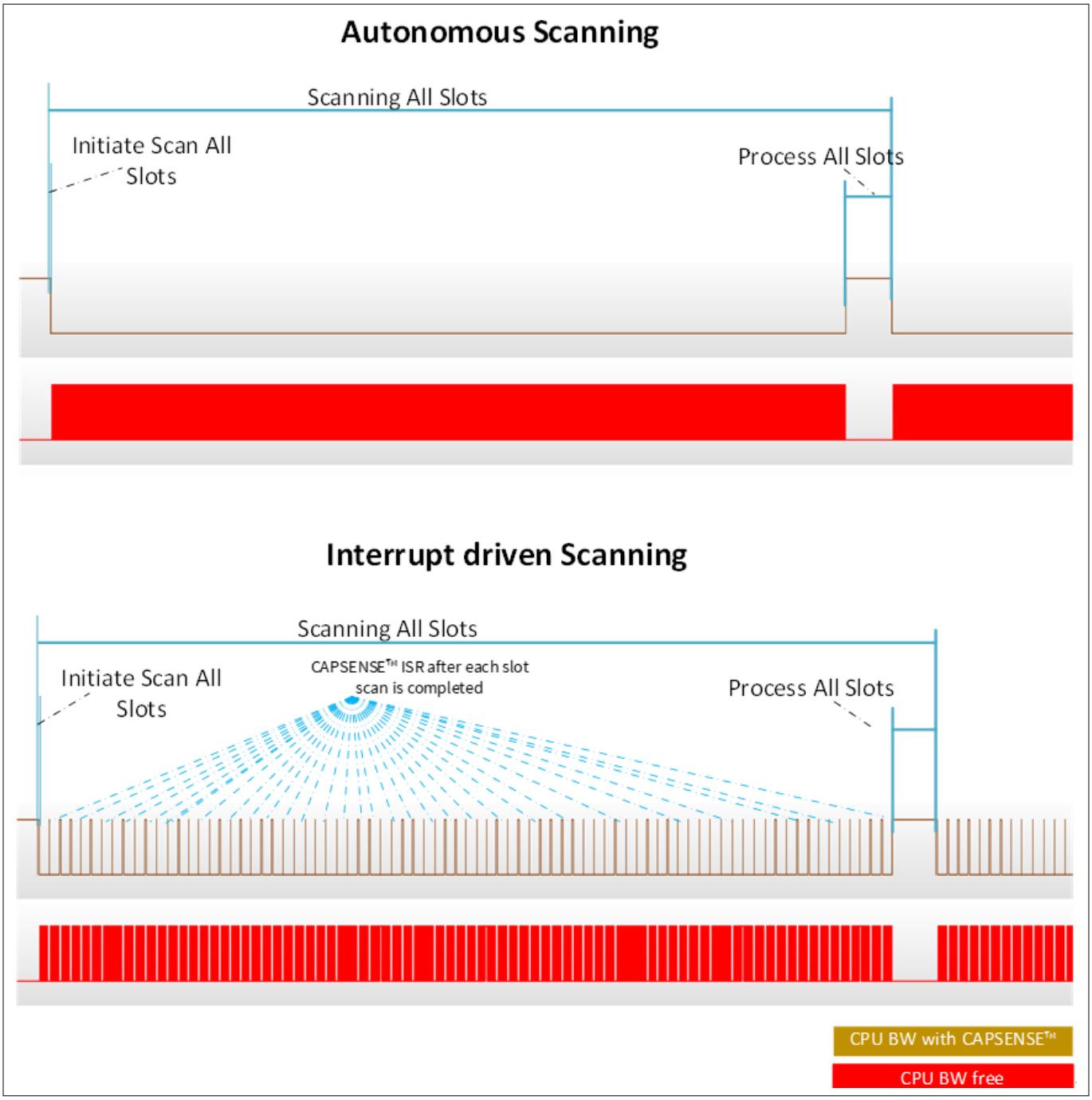


図 92　　自律スキャンと割込み駆動スキャンの比較

第 5 世代 CAPSENSE™では、自律スキャンは、図 93 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 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、スキャン結果を読み出し、次のセンサーを構成するためには、CPU 割込みを処理する必要があります。

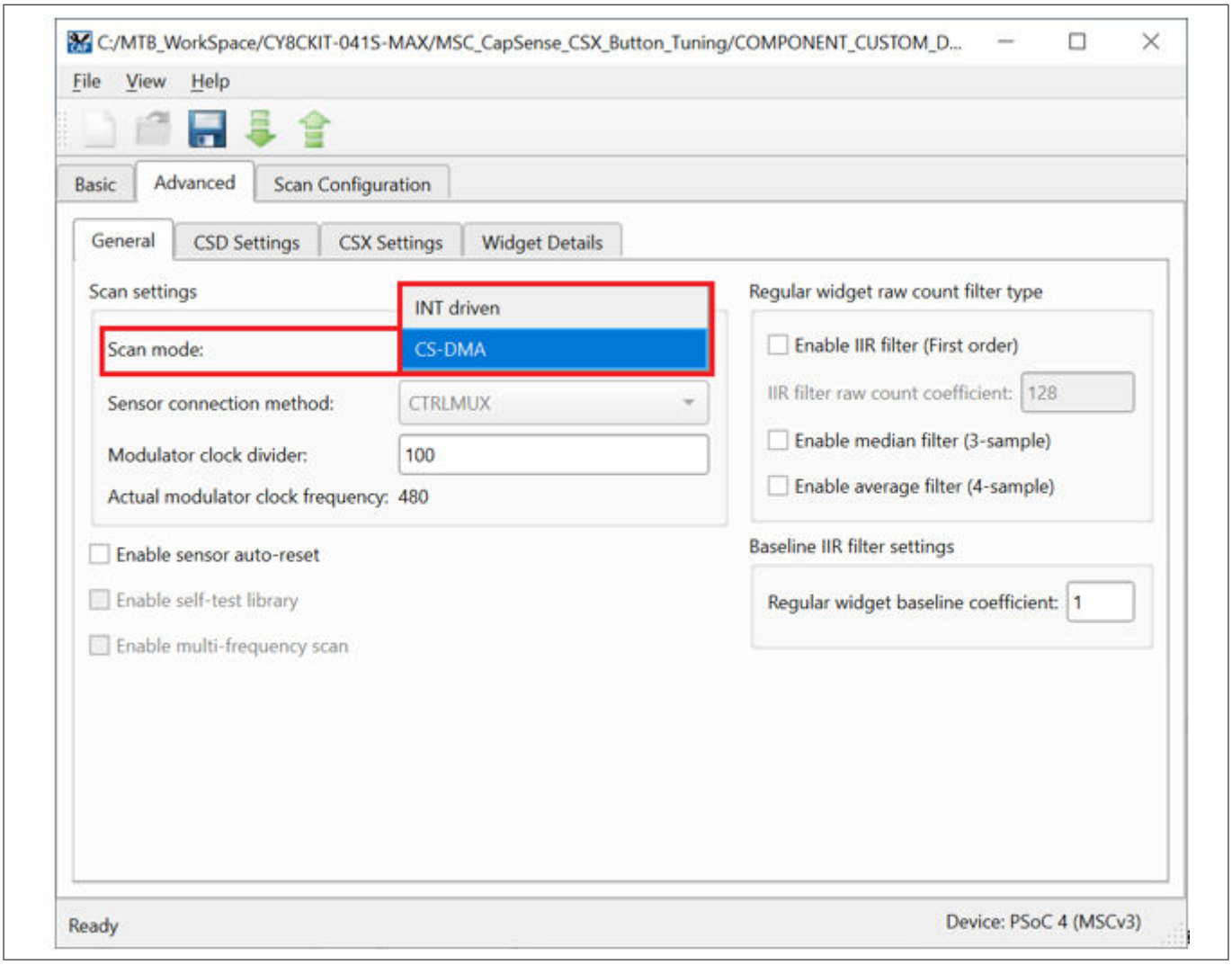


図 93　　スキャンモードの CAPSENSE™コンフィギュレーター設定

第 5 世代の低消費電力 CAPSENSE™では、通常のウィジェットはデフォルトで自律的にスキャンされます。スキャンが行われている間、CPU はディープスリープモードのままにしておくことができます。割り込みは、すべてのスロットがスキャンされた後、データを処理するために CPU をウェイクアップさせることができます。低電力ウィジェットは、LP-AoS モードで説明されているように、低電力常時オンセンス (LP-AoS) モードを使用してスキャンされます。

自律スキャン

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。CAPSENSE™世代に基づく自律スキャンには、連鎖スキャン– DMA モードと LP-AoS モードの方法があります。

連鎖スキャン– DMA モード

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。図 94 に示すように、MSC ブロックの各チャネルでは、デバイスコンフィギュレータで 4 つの DMA チャネルを設定する必要があります。

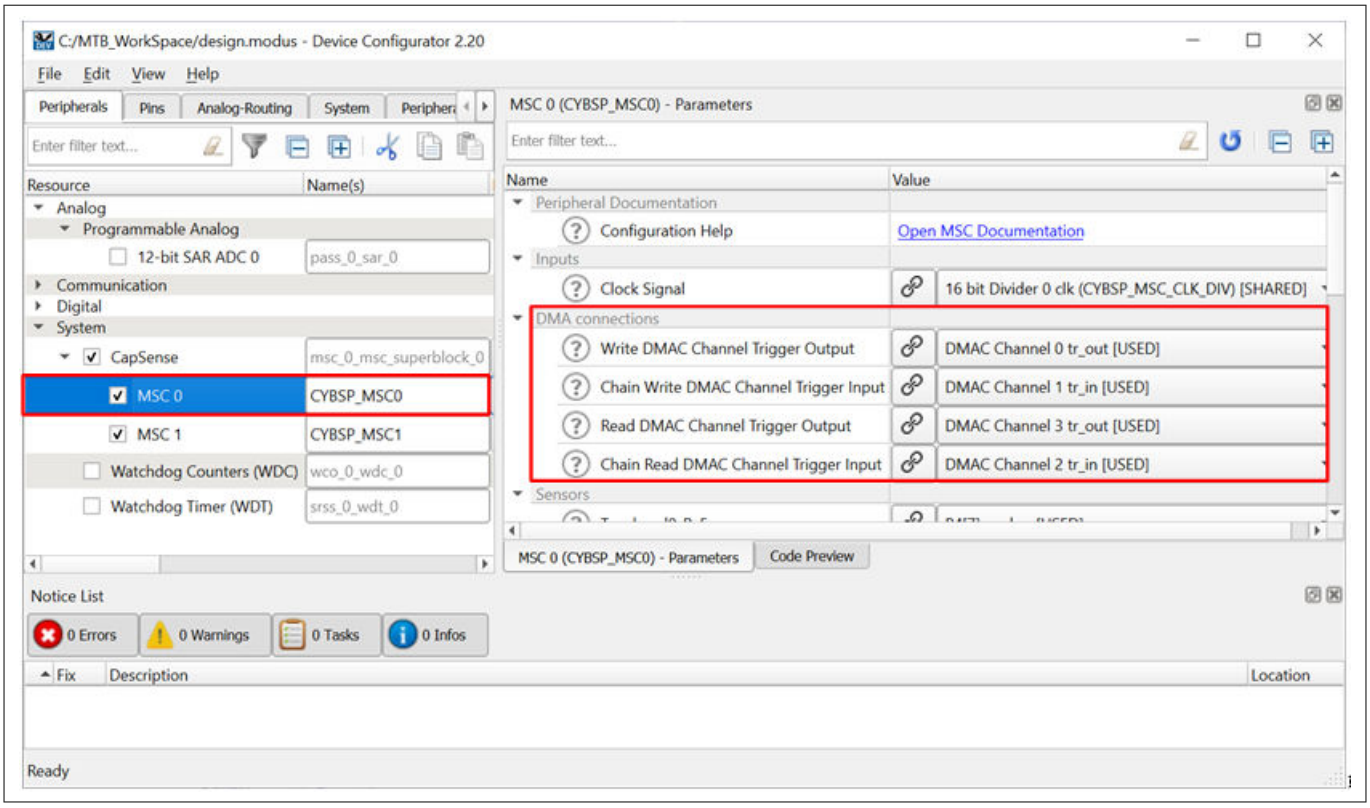


図 94　　MSC チャネルの DMA 構成

図 95 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 チャネルから受信されます。

2. Chain Write DMA チャネル

   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 チャネルに転送されます。

3. Read DMA チャネル

   Read DMA チャネルは、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 チャネルに転送されます。

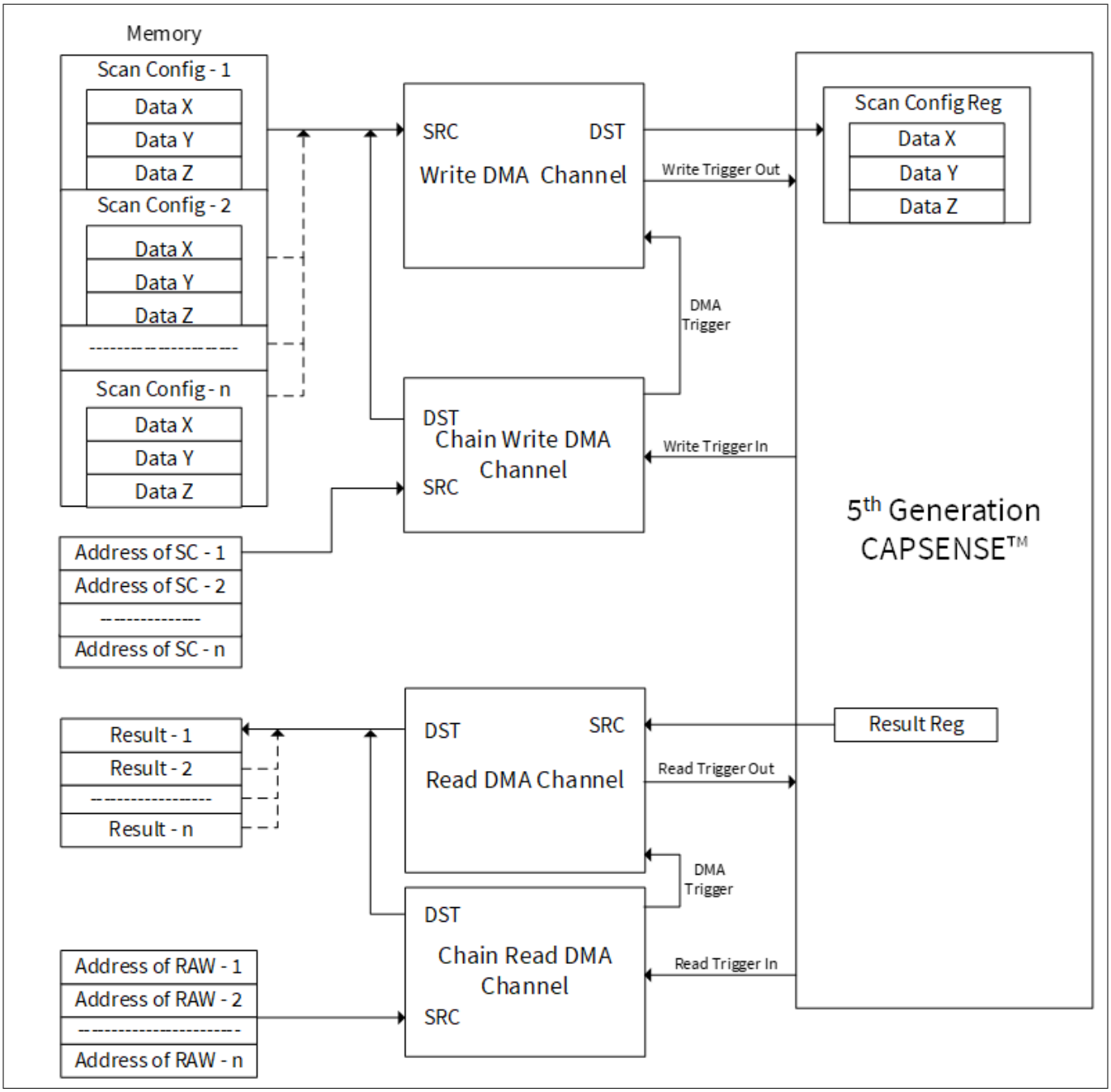


図 95　　CS-DMA スキャンニングフロー

LP-AoS モード

第 5 世代の CAPSENSE™ (MSCLP) は、LP-AoS ハードウェアモードを使って自律センシングを行います。このモードを使用すると、CPU を介さずに複数のセンサー (フレーム) をスキャンできます。これを可能にするためには、MSCLP ブロックは 1KB の SRAM を内蔵している必要があります。ミドルウェアは、スキャンする複数のセンサーの設定を内蔵 SRAM に書き込みます。そして、MSCLP は自律的に各センサーの設定を読み取り、スキャンを開始し、その結果を SRAM に保存して、フルフレームスキャンが完了した後に、ミドルウェアが読み取って処理します。

MSCLP が複数のセンサーをスキャンしている間、デバイスはアクティブ、スリープ、ディープスリープなど、どのような電力モードにも保つことができます。MSCLP は、デバイスの低消費電力状態 (すなわちディープ スリープ) で

のスキャンを可能にするため、高周波クロックや基準発生器などのシステム リソースをすべて MSCLP ハードウェアに内蔵しています。

ウェイクオンタッチ (WoT)

LP-AoS には、同じフレームを定期的にスキャンし、その結果を処理してタッチを検出する機能が追加されています。そのため、CPU はスキャンを処理または開始する必要がありません。これにより、PSOC™デバイスをより長い時間ディープスリープ状態に保ち、センサーのタッチ検出をウェイクアップ割り込みとして使用することができます。これはウェイクオンタッチとして知られています。このアプローチにより、アプリケーションはタッチ センシング機能で最も低い消費電力を得ることができます。WoT は低電力ウィジェットでのみサポートされていることに注意してください。低電力ウィジェットの詳細については、低消費電力ウィジェット チューニングを参照してください。

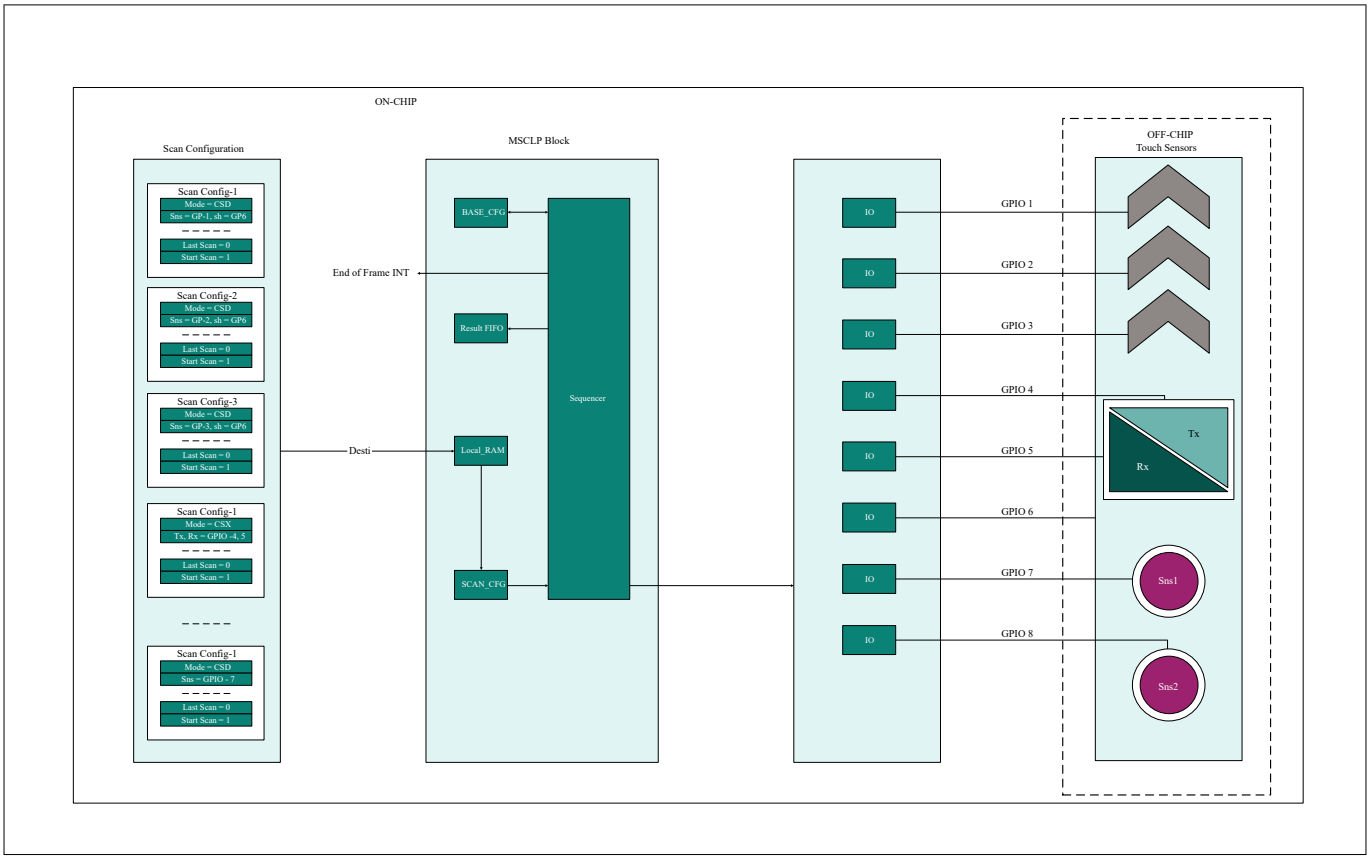


図 96　　LP-AoS モード

5.3.4.2.6　　マルチチャネルスキャン

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 タッチパッドは図 97 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ネル 0 (MSC0) でスキャンされ、緑色はチャネル 1 (MSC1) でスキャンされます。

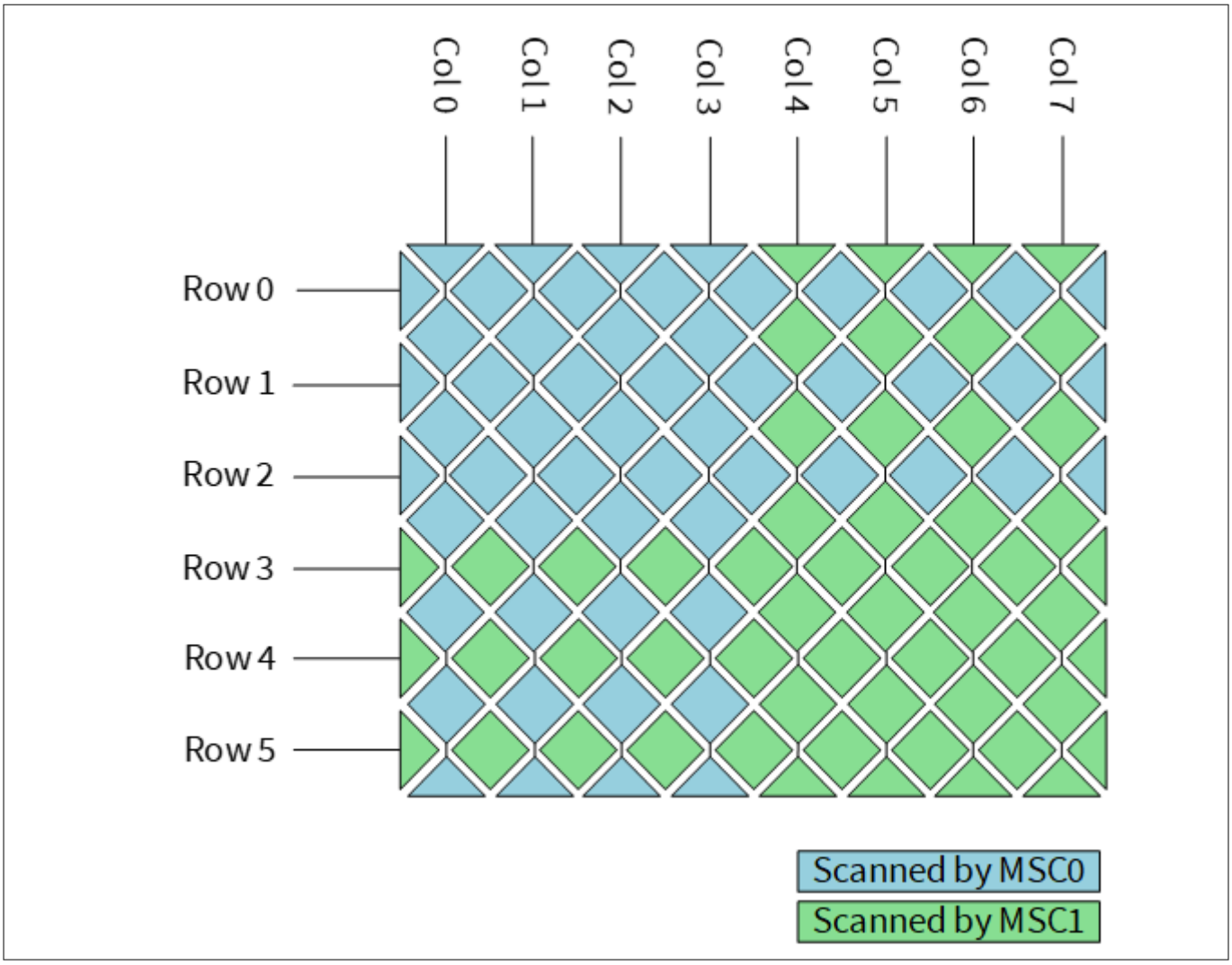


図 97　　マルチチャネルを使用した 6x8CSD タッチパッドのスキャン

この場合、チャネル 0 とチャネル 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表 20 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。

表 20　　チャネルスキャン設定

スロット#	チャネル 0 センサー	チャネル 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

5.3.4.2.7 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。

5.3.4.2.8 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。

5.3.4.2.9 タッチパッドウィジェットの調整

タッチパッドのハードウェアとファームウェアの設計ガイドラインについては、チューニングガイドラインとともに AN234185 PSOC™ 4 CAPSENSE™ Touchpad design guide を参照してください。CE235338 PSOC™ 4: MSCLP self-capacitance タッチパッド tuning サンプルコードは、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。

PSOC™ Creatorb ベースの CAPSENSE™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。

5.3.4.2.10 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 - CAPSENSE™による近接センシングを参照してください。

5.3.4.2.11 低消費電力ウィジェット チューニング

CE235111 - PSOC™ 4: MSCLP CAPSENSE™ low power CSD Button Tuning では、ModusToolbox™用 Eclipse IDE における低消費電力ウィジェットのチューニングについて説明します。

5.3.5 CSX-RM センシング方式 (第 5 世代および第 5 世代低消費電力)

ここでは、第 5 世代および第 5 世代低消費電力デバイスの 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、表 10 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 66 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。

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

式 66　　raw カウントとセンサー静電容量の関係

ここで、

G_{CSX} = CAPSENSE™ CSX の容量からデジタル変換利得

C_M = Tx および Rx 電極間の相互容量

式 38 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 67 の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。

この変換利得の概算値は次のとおりです。

$$G_{CSX} = MaxCount\frac{2}{C_{ref}TxClk_{Div}}$$

式 67　　静電容量-デジタル変換利得

ここで、MaxCount = N_{Sub} * $TxClk_{Div}$

式 66 および式 67 による、シングル CDAC モードでの raw カウントの式を式 68 に示します。

$$Rawcount_{Counter} = N_{Sub}\frac{2 \times C_M}{C_{ref}}$$

式 68　　シングル CDAC モードでの raw カウント

ここで、

N_{Sub} = サブ変換回数

$TxClk_{Div}$ = Tx クロック分周器

C_M = センサーの相互容量

RefCDACCode = 基準 CDAC 値

C_{lsb} = 8.86 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。

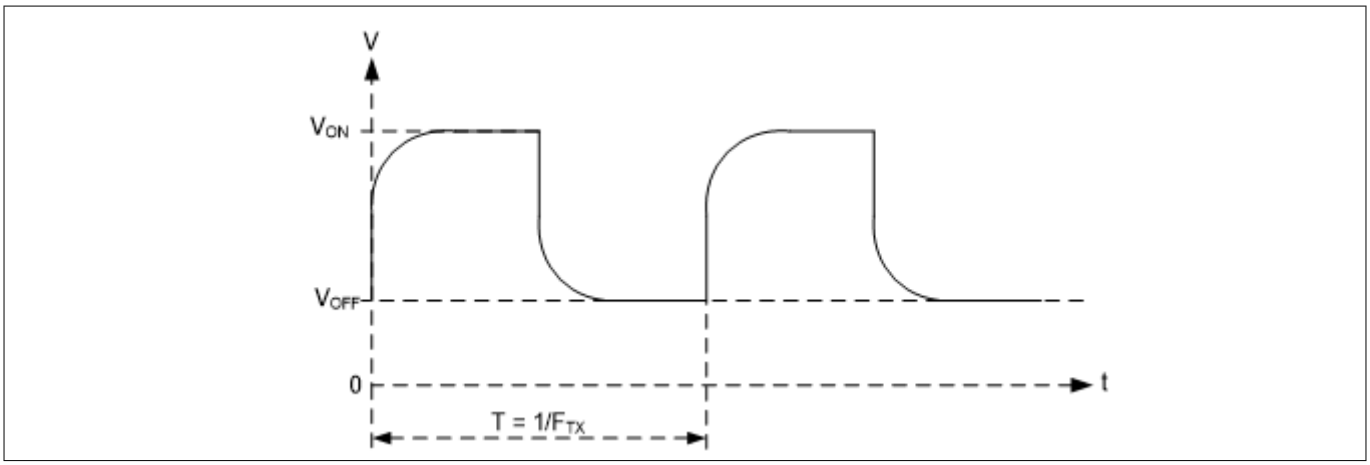


図 98　　CSX センサーの Tx ノードでの電圧

注: *コンポーネントから観測された raw カウントは式 69 で与えられます。$Rawcount_{component}$ の詳細については、CAPSENSE™ CSX-RM センシング方式 (第 5 世代および第 5 世代低消費電力)を参照してください。*

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

式 69 $Rawcount_{component}$

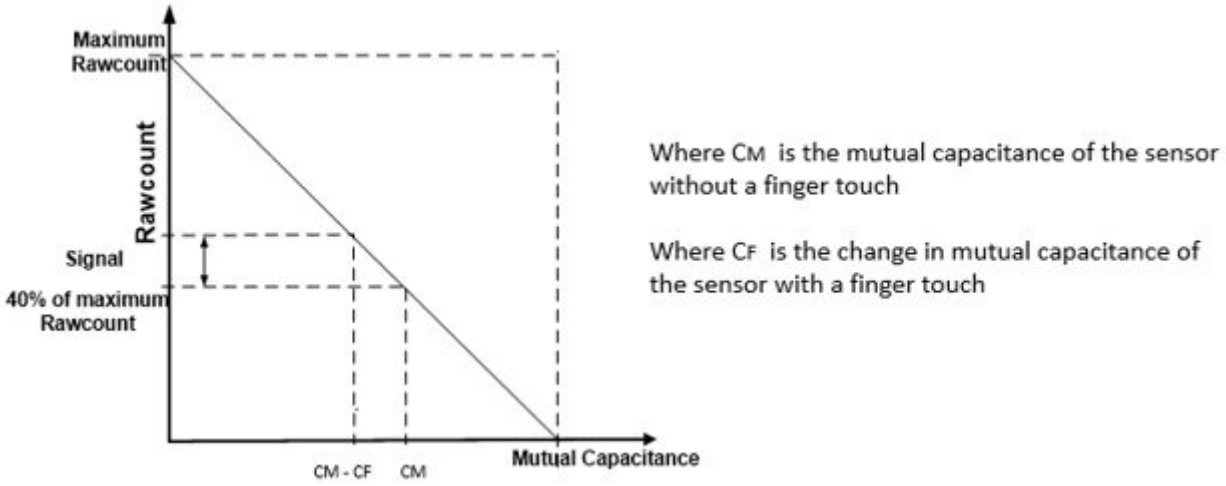


図 99 raw カウントとセンサーの相互容量

デュアル CDAC モードでの変換利得

式 27 および式 66 による、デュアル CDAC モードでの raw カウントの式を式 70 に示します。

$$\text{rawcount} = G_{CSX}C_M - Maxcount\frac{2 \times C_{comp}}{C_{ref}CompCLK_{div}}$$

式 70 デュアル CDAC モードでの raw カウント

ここで、

Maxcount = N_{Sub} *$SnsClk_{Div}$

$SnsClk_{Div}$ = センス クロック分周器

N_{Sub} = サブ変換回数

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

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

$CompCLK_{Div}$ = 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_{lsb} = 8.86 fF

G_{CSX} は式 67 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。表 21 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。

表 21 CAPSENSE™コンポーネント ハードウェア パラメーター

S. 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	Fine CDAC 値
8	サブ変換回数
9	CDAC ディザを有効

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。

第 5 世代の低消費電力 CAPSENSE™では、自律スキャンが本質的にサポートされており、CPU の介入を必要とせず、CTRLMUX と DMA を必要としません。

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 (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、この方式によってオンチップ I/O 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。

第 5 世代の CAPSENSE™テクノロジでは、CTRLMUX によって限られたピン数がサポートされていましたが、第 5 世代低消費電力 CAPSENSE™では、すべての CAPSENSE™ピンが AMUXBUS 経由で自律スキャンモードをサポートします。

5.3.5.2.3 変調器クロック周波数

式 66、式 67 に示すように、変調器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。

5.3.5.2.4 初期化サブ変換

初期化の一環として、C_{MOD} 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} short, 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、特定の初期化サブ変換数のために raw カウントは破棄されます。

式 71 に基づいて初期化サブ変換数を選択してください。

$$\text{Number of init subconversions} = ceiling\left(\frac{C_{MOD} \times V_{OS}}{2 \times VDDA \times C_M \times \left(1 - Base\,\%\right) \times \left(\frac{1}{Bal\,\%} - 1\right)}\right) + 1$$

または

$$\text{Number of inti subconversions} = ceiling\left(\frac{C_{MOD} \times V_{OS}}{VDDA \times TxClkDiv \times C_{ref} \times \left(1 - Bal\,\%\right)}\right) + 1$$

式 71 初期化サブ変換数

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)

***注**: 車載グレード (E グレード) のデバイスには Vos=4mV を使用してください。*

C_M = センサーの相互容量

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

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

RefCDACCode = 基準 CDAC 値

C_{lsb} = 8.86 fF

***注**: CSX の場合、"Enable coarse initialization bypass "パラメータはチェックしないことを推奨します。*

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。

- 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 (PRSx) – 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、PRS 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。

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。

PRS/SSC はエミッションを低減するために使用できますが、高い Nsubs を必要とするため消費電力が増加します。最大拡散のためには、Nsubs を 1024 (または 1024 の倍数) に設定することを推奨します。Nsubs が 1024 より小さい、または 1024 の倍数でない場合、パフォーマンスは最適化されません。

Tx クロック周波数

CAPSENSE™ CSX-RM センシング方式 (第 5 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 クロック分周は、センサー容量を完全に充放電させるために、Tx クロックのパルス幅が十分な長さになるように構成してください。これは、オシロスコープとアクティブプローブを用いてセンサーの充放電波形を観察することで検証できます。センサーの測定は、センス ピンや直列抵抗ではなく、電極の近くで行ってください。

センサー/シールドで観測された波形については、図 100 および図 101 を参照してください。図 100 は、センスクロック周波数が正しく調整されている場合、適切な充電を示します。パルス幅は少なくとも 5Tau です。つまり各フェーズの終了時に電圧は必要電圧の少なくとも 99.3%に達しています。図 101 は、不完全な整定 (充放電) を示します。充電が不完全な場合は、Tx クロック分周を増やしてください。

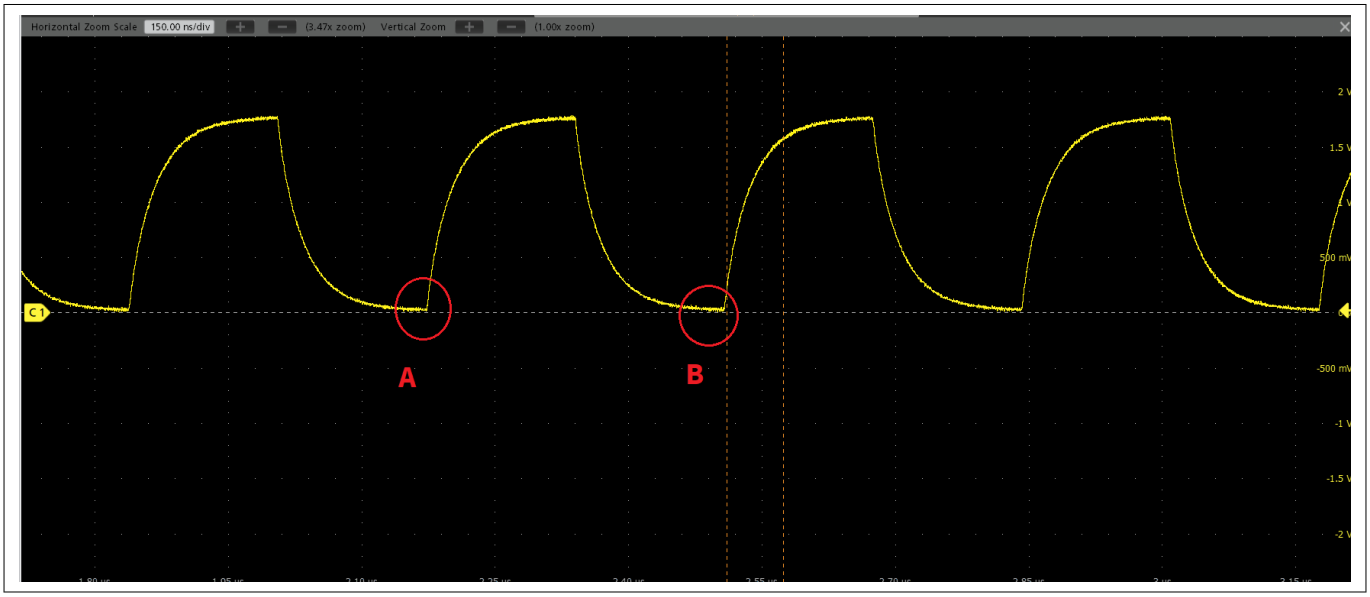


図 100　　CSX センサーの適切な充電サイクル

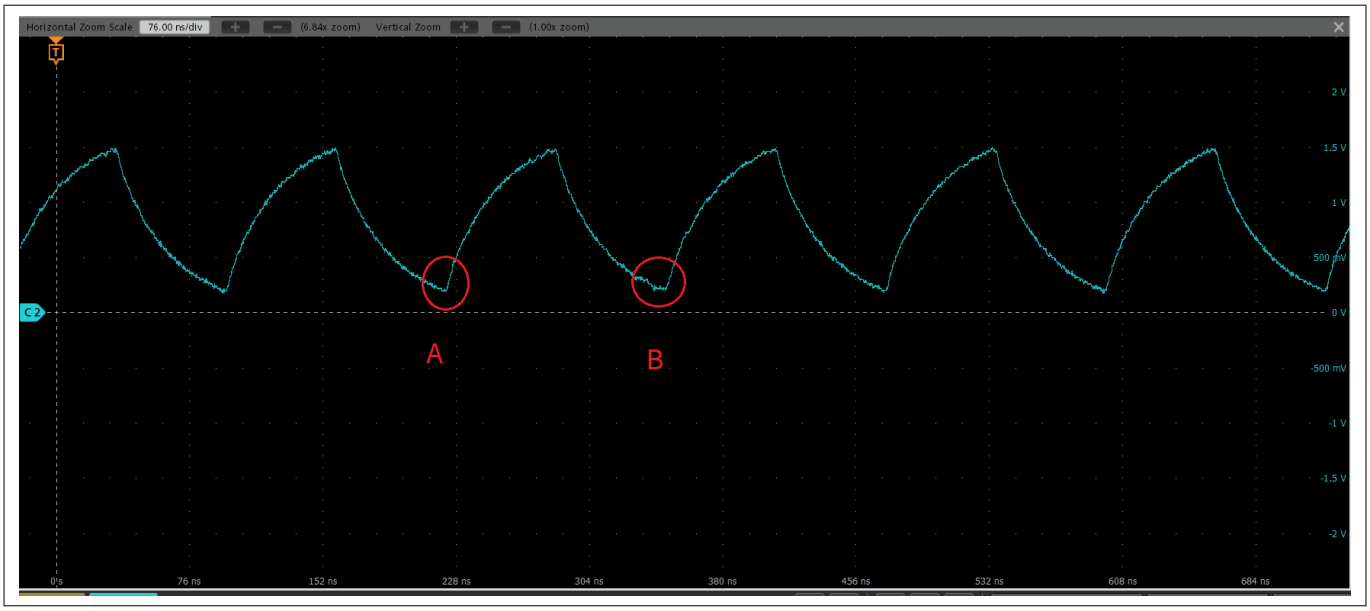


図 101　　CSX センサーの不適切な充電サイクル

***注**:　センスクロック分周器は **2 で割り切れる**必要があります。これにより、両方のスキャンフェーズの時間が同じになります。*

全てのセンサーとシールドに対して同じ手順を繰り返してください。各センサが完全に充放電するためには、異なる Tx クロック分周値が必要な場合があります。しかし、同じスキャンスロットにあるすべてのセンサーは、同じ Tx クロックソース、Tx クロック分周器、サブコンバージョン数を持つ必要があります。したがって、与えられたスキャンスロットで最大の Tx クロック分周を取り、そのスロットを共有する他のすべてのセンサーに適用してください。

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/ミドルウェアドキュメント/ 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　サブ変換回数

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 27 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、サブ変換数を増やすと、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段階的に N_{Sub} を増やしてください。

***注**:　CDAC ディザリングが最適に機能するように、サブコンバージョン数は 8 以上でなければいけません。PRS/SSC を使用する場合は、最適なパフォーマンスを得るために 1024 の倍数にする必要があります。*

5.3.5.2.7　容量性 DAC

第 5 世代の CSX-RM は、式 8 に示すように、C_{MOD} のバランスをとる基準 CDAC (C_{ref})、Fine CDAC (Cfine)、および補償 CDAC (C_{comp}) の 3 つの CDAC 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変換利得および 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

表 22 は CAPSENSE™コンフィギュレータで設定できる有効な CDAC の組み合わせを示します。

表 22　　有効な CDAC 構成

Sr. no.	C_{ref}	C_{fine}	C_{comp}
1	自動	自動	自動
2	自動	自動	無効
3	自動	無効	自動
4	自動	無効	無効
5	設定値	設定値	自動
6	設定値	設定値	設定値
7	設定値	設定値	無効
8	設定値	無効	自動
9	設定値	無効	設定値
10	設定値	無効	無効
11	無効	自動	自動
12	無効	自動	無効
13	無効	設定値	自動
14	無効	設定値	設定値
15	無効	設定値	無効

***注**:　　CDAC の場合、自動キャリブレーションを常に有効にすることを推奨します。これにより、最適なチューニングを確保できます。*

基準 CDAC (C_{ref})

基準 CDAC は、C_{MOD} からセンサーの相互容量 (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、以下の方程式を使用して計算します。

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

ここで、

RefCDACCode = 基準 CDAC 値

C_{lsb} = 8.86fF

C_{ref} は以下の基準を満たす必要があります。

$$\text{RefCDACCode} > 6$$

自動校正された RefCDACCode が 6 より小さい場合は、Fine CDAC を有効にします。

RefCDACCode を増やすためには、以下のオプションを使用できます。

- センスクロック分周を下げてください。センスクロックのパルス幅は、センサー容量が完全に充放電するのに十分な長さでなければいけないことに注意してください。このセンスクロックの低下は、直列抵抗を小さくするか、シールド Cp を小さくすることで対応できます。
- 補正を無効にしてください

Fine 基準 CDAC (C_{fine})

第 5 世代の低消費電力 CAPSENSE™は、Fine CDAC で構成されます。Fine CDAC はプログラマブル CDAC で、基準 CDAC のより微細な分解能を達成するために使用されます。したがって、Cfine を Cref とともに有効にすることで、精度が向上します。Fine CDAC の場合、自動キャリブレーションを常に有効にすることを推奨します。これにより最適なパフォーマンスが保証されます。

C_{fine} = Fine 基準 CDAC = FineCDACCode * C_{lsb}

FineCDACCode = Fine CDAC 値

C_{lsb} = 2.2 nF

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、変換利得および 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。

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 72 に基づいて CDAC 補償分周器を選択します。

1. CDAC 補償分周器 > = 4
2. K_{comp} は整数である必要があります。

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

式 72 CDAC 補償分周器

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 を自動的に調整して、必要な調整目標である 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

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 66, 式 67, および式 69 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™コンポーネントを構成し、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構成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。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。

最適なパフォーマンスを確保するため、常に自動キャリブレーションモードで CDAC ディザスケールを有効にすることを推奨します。CDAC ディザスケールを手動で設定する必要がある場合は、以下の表で一般的な推奨値を確認してください。

表 23 CSX センサーのディザスケール推奨値

相互キャパシタンス (C_m)	スケール
300 fF <= C_m < 500 fF	5
500 fF <= C_m < 1000 fF	4
1000 fF <= C_m < 2000 fF	3
C_m >= 2 pF	表 19 に準拠

ディザ関連パラメータの設定の詳細については、PSOC™ 4: MSCLP CAPSENSE™ low power (section '**Stage 2: Measure sensor capacitance to set CDAC Dither parameter**')を参照してください。

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 自律スキャンの設定

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

5.3.5.5 マルチチャネルスキャン

CSX-RM センシングのマルチチャネルスキャンは、CSD-RM センシングのマルチチャネルスキャンと同じです。したがって、詳細については、第 5.3.4.2.6 章セクションを参照してください。

5.3.5.6 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、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。

5.3.5.7 タッチパッドウィジェットの調整

タッチパッドのチューニングガイドラインについては、AN234185 PSOC™ 4 CAPSENSE™ Touchpad design guide を参照してください。

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ことが必要です。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、管理することが必要です。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。オーブンの「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。図 102 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。

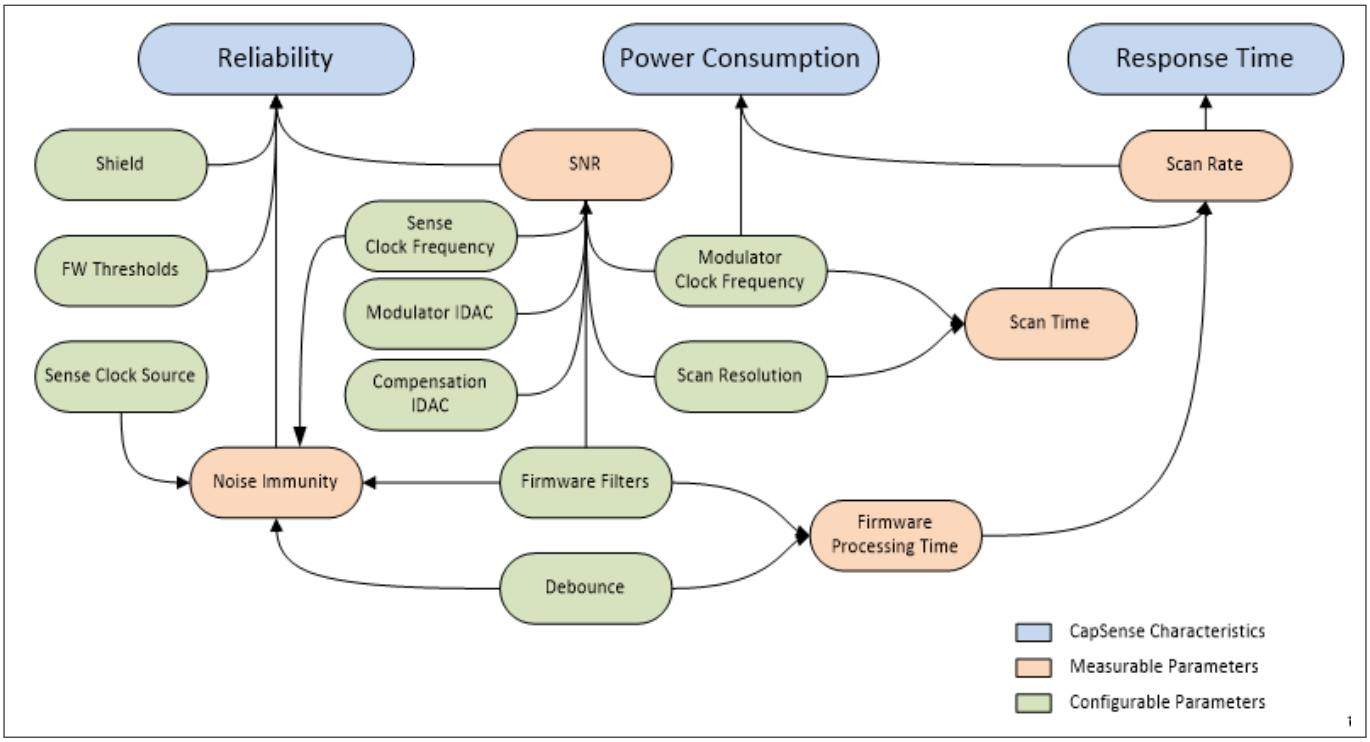


図 102　　CAPSENSE™パラメーターの関係

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。

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。チューナーの使用方法の詳細については 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。

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。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 ドキュメント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。

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 および補正 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。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 ドキュメントを参照してください。

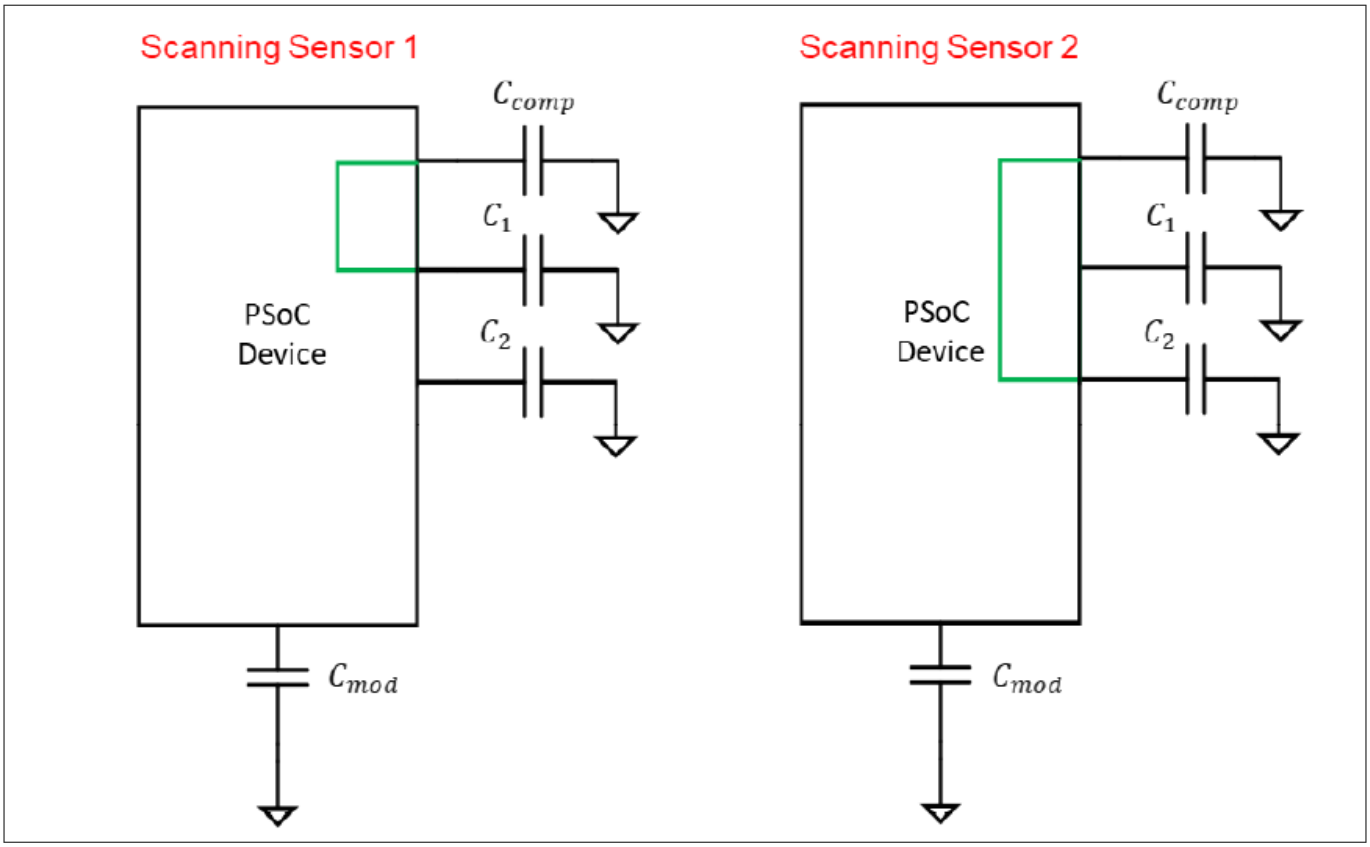


図 103　　センサーを外部補償コンデンサに接続

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 の詳細についてはコンポーネント データシート/ミドルウェア ドキュメント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 CP 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確認し、必要に応じてレイアウトを更新してください。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);
```

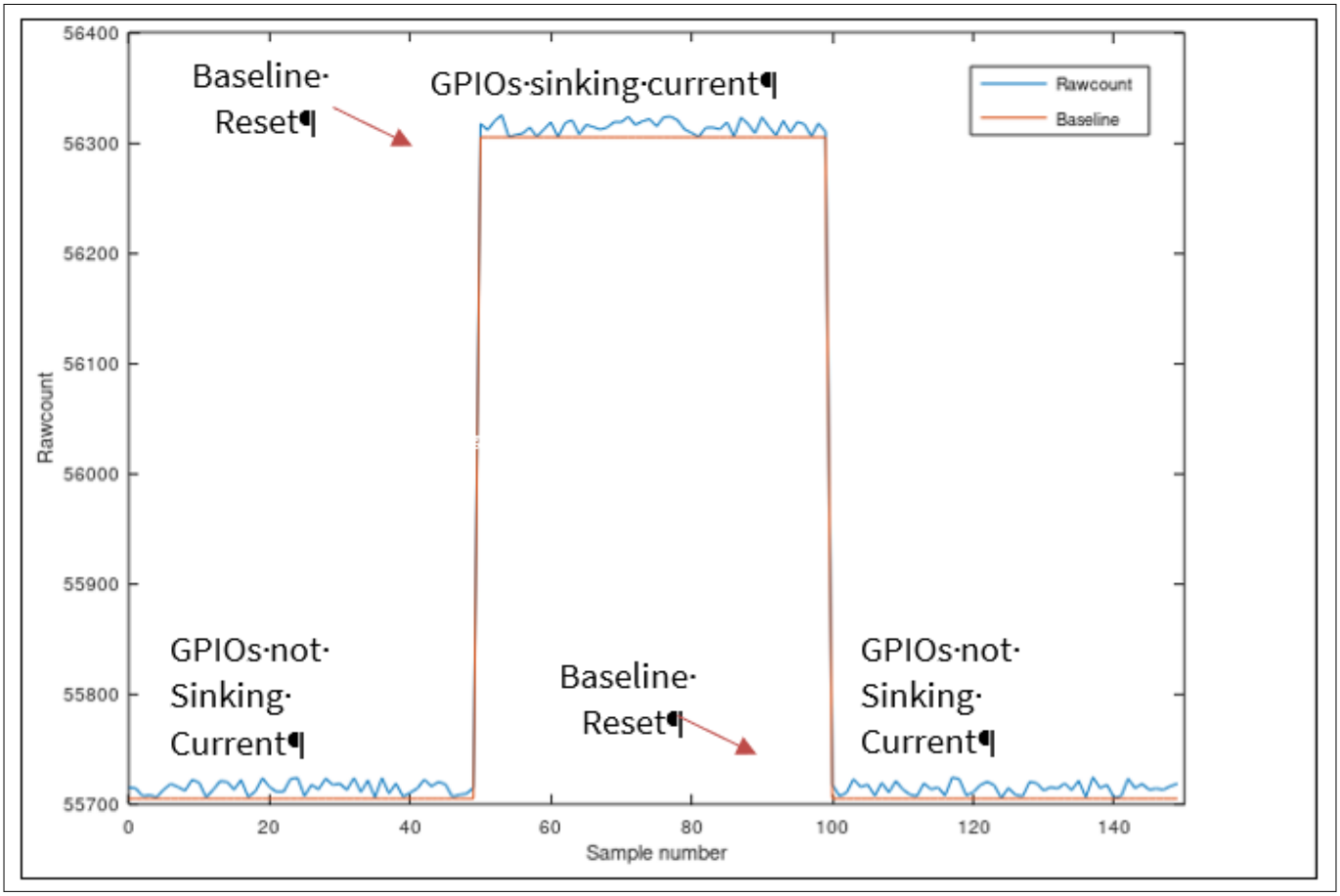


図 104　　ファームウェア技術を使用したベースラインのリセット

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。

- スイッチング GPIO のドライブ強度を下げてください。表 24 に、GPIO で使用可能なドライブ強度オプションを示します。図 105 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。

表 24　　GPIO の駆動強度

駆動強度	駆動電流 (mA)
Full	8

(続く)

表 24 (続き) GPIO の駆動強度

駆動強度	駆動電流 (mA)
$\frac{1}{2}$	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。
- 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、表 42 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。

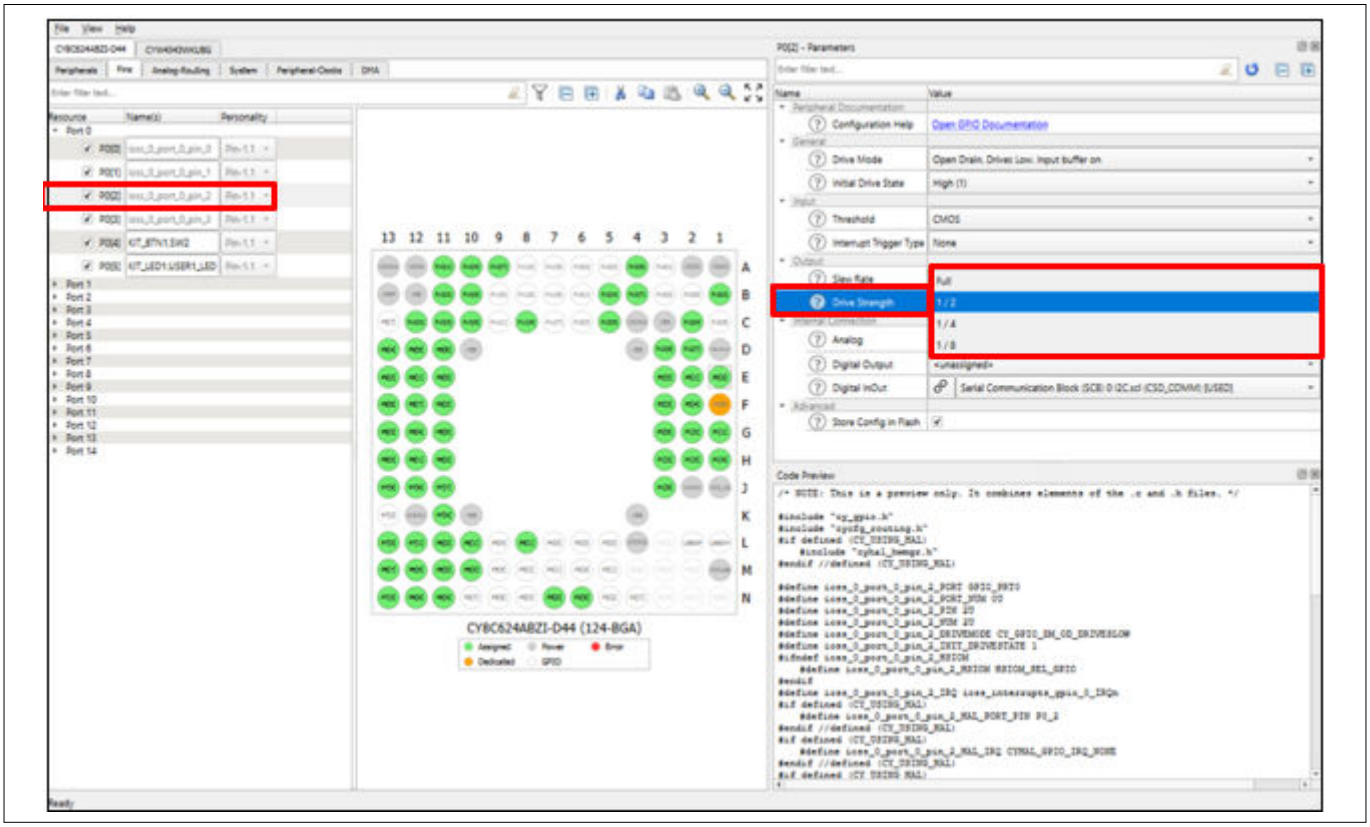


図 105 GPIO の駆動強度の選択

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 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。図 106 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 を参照してください。

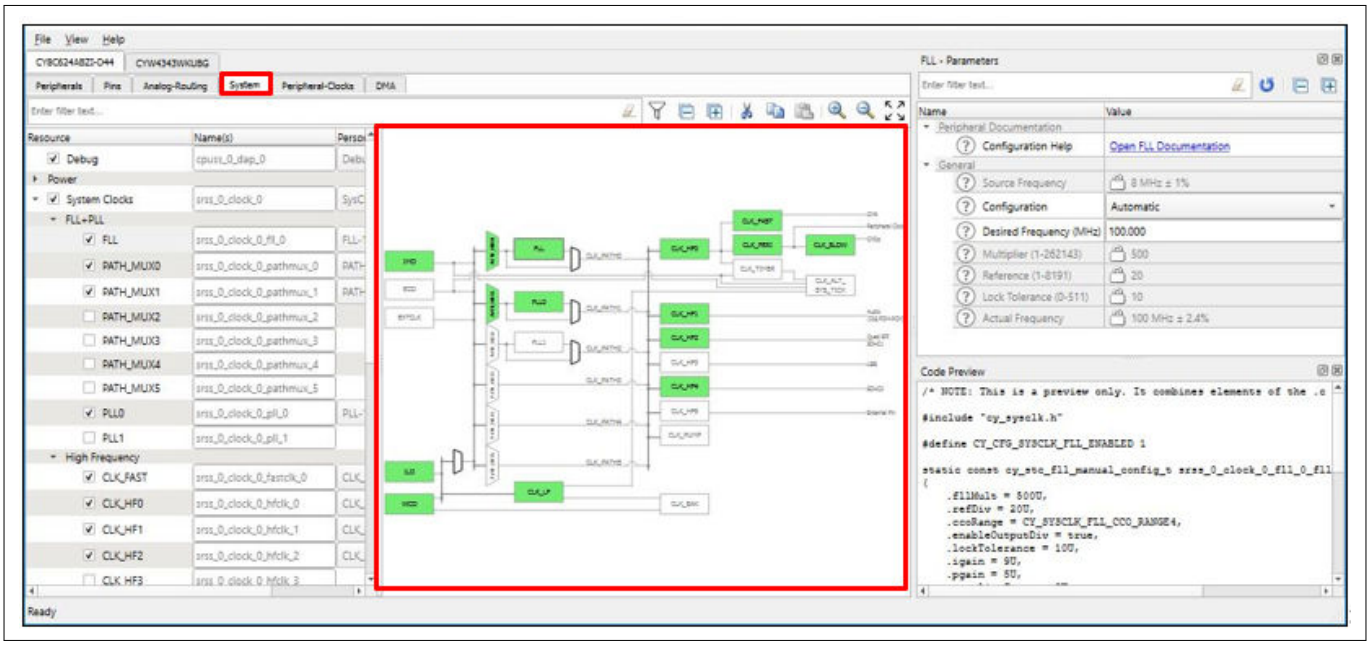


図 106 Device configurator でのクロック設定の変更

5.3.7.12 CSX ボタンの CM 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 ドキュメントを参照してください。

図 107 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。図 108 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。

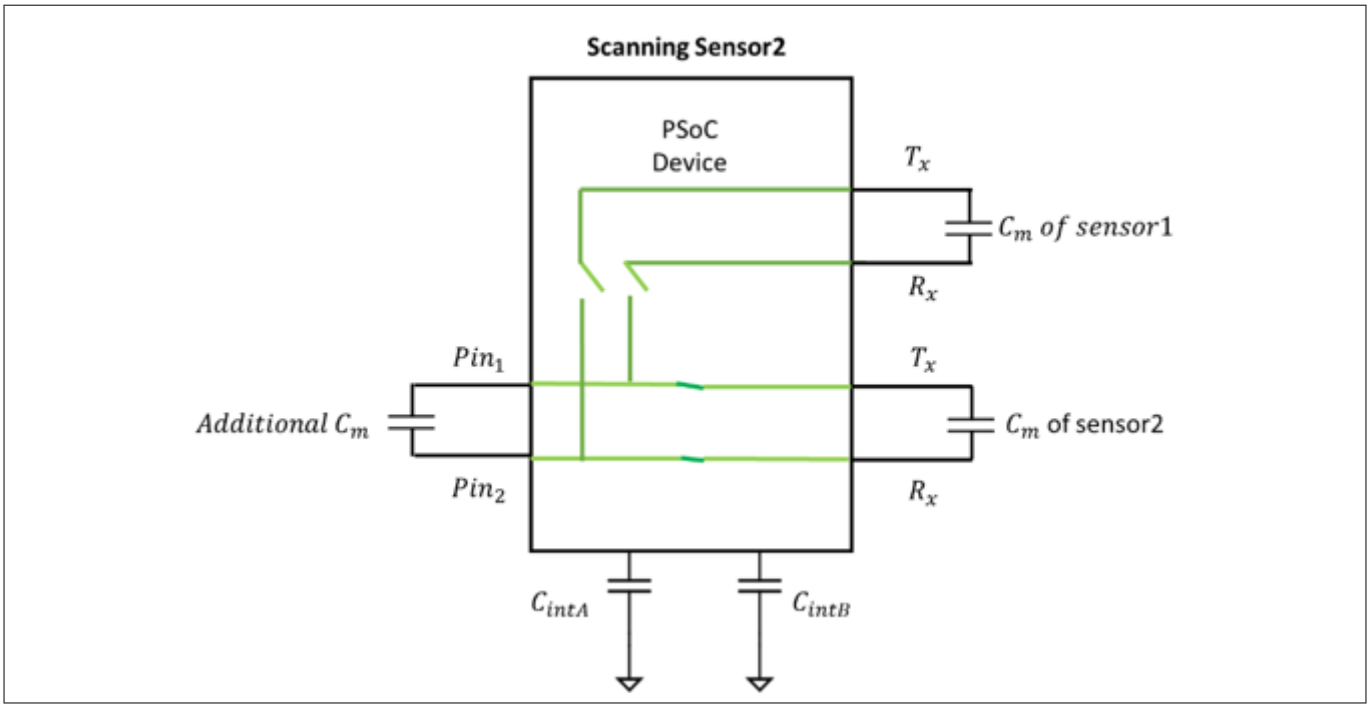


図 107　　センサーの C_M を増加させるための外部コンデンサのギャング

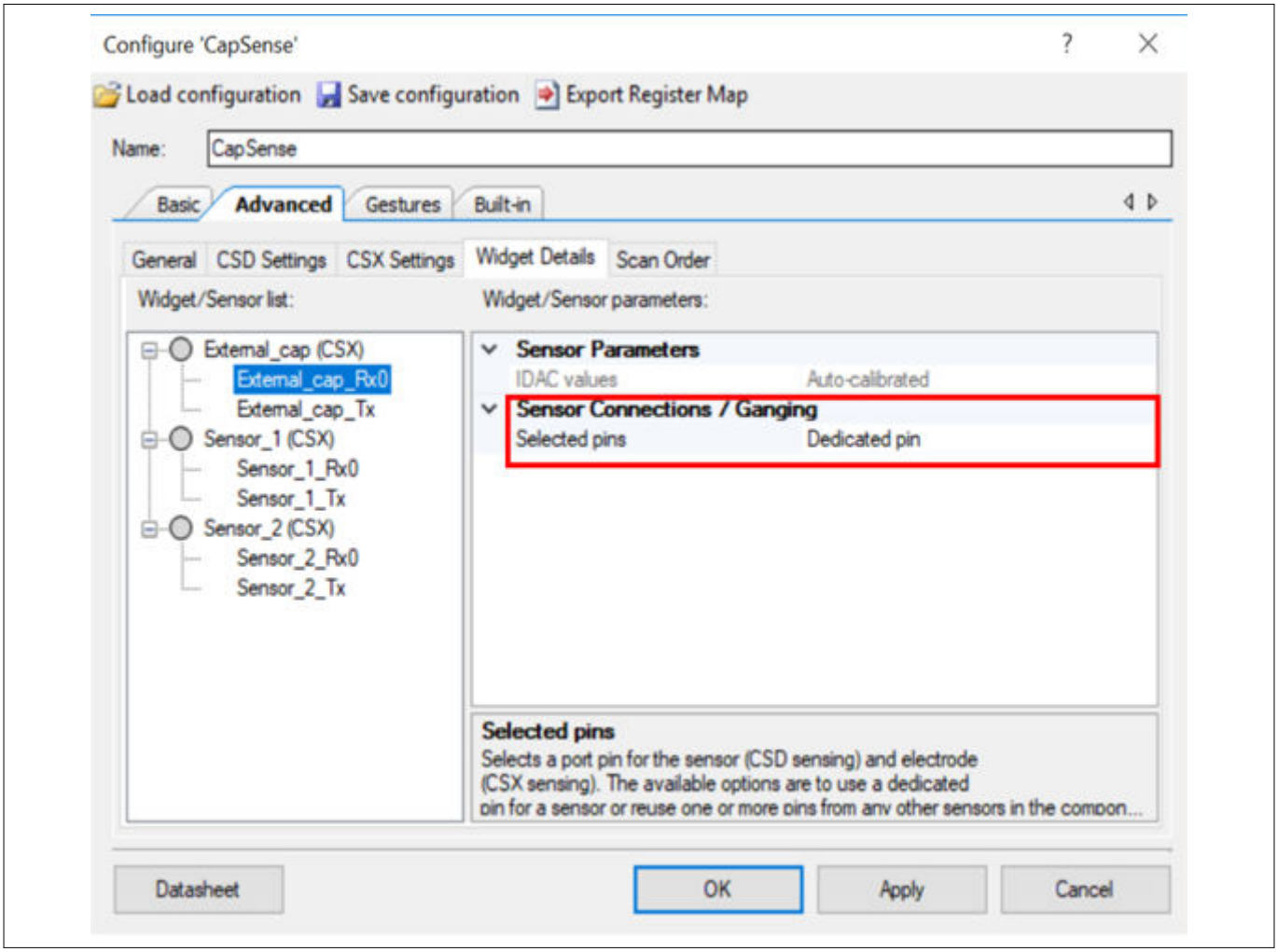


図 108　　外部コンデンサへの専用ピンの割り当て

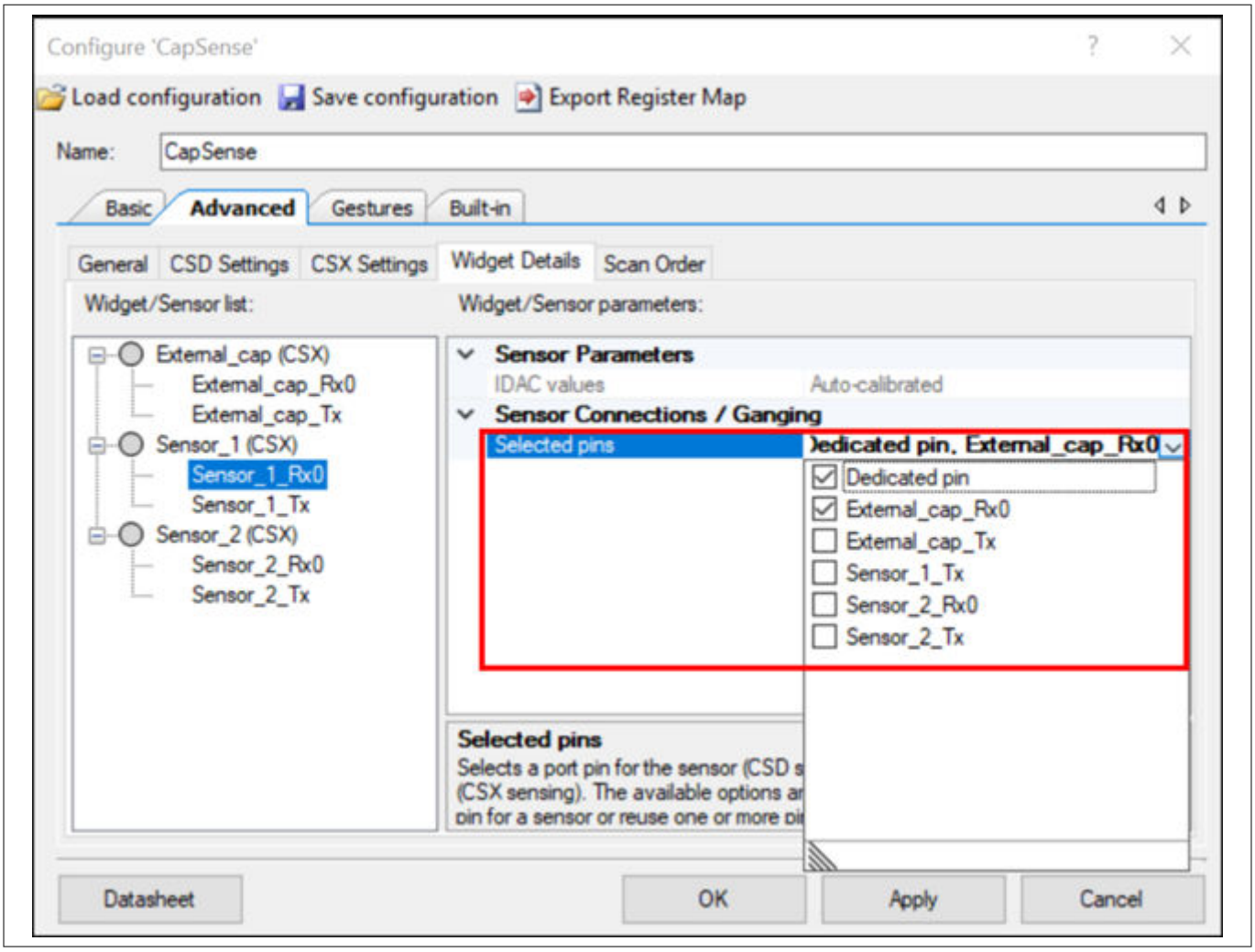


図 109　　外部コンデンサとセンサーピンの取得

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。

注: 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ます。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。

表 25 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 ドキュメントを参照してください。

表 25 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、表 25 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す。
- **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 ドキュメントを参照してください。

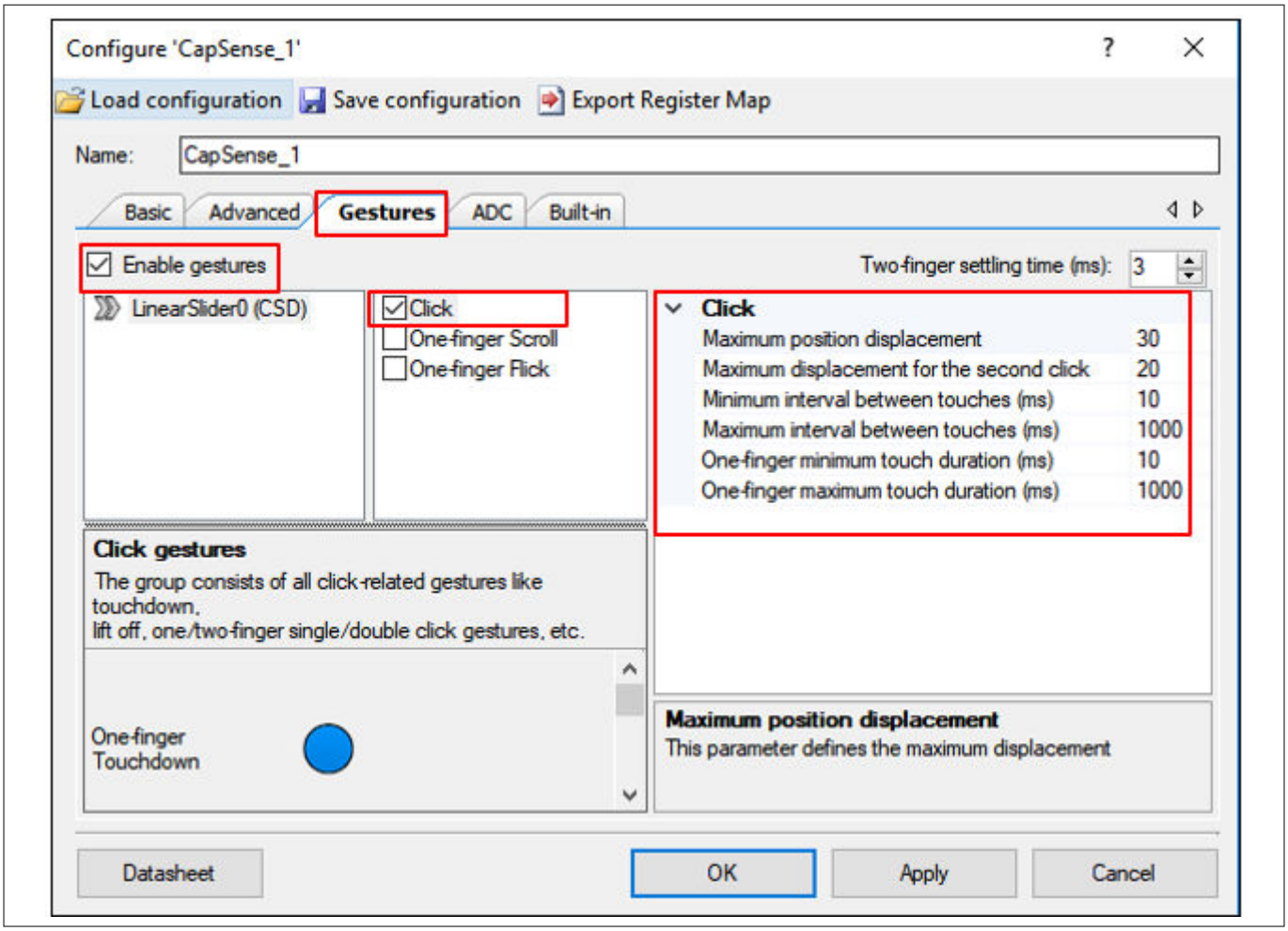


図 110　　CAPSENSE™コンポーネントのジェスチャー設定

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、スライダー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™スライダーは、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/ミドルウェアドキュメント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 ドキュメントおよび 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. 図 111 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。

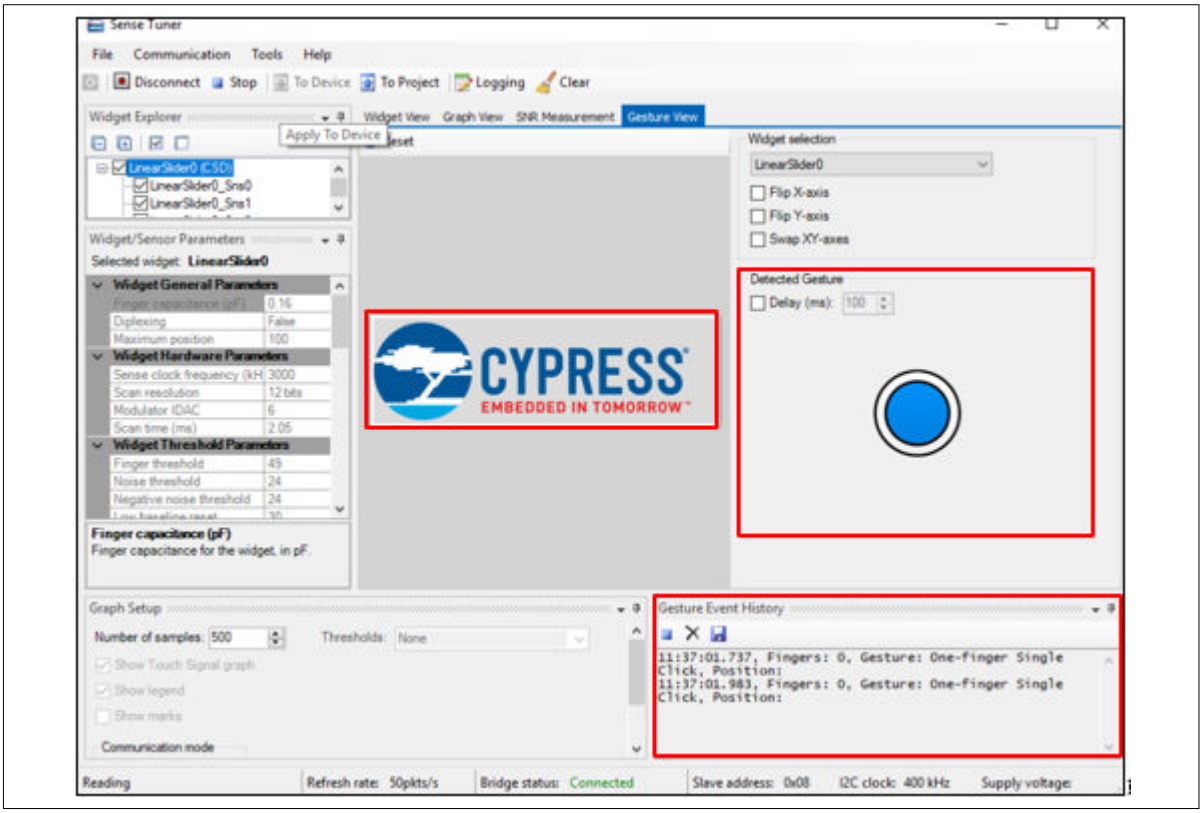


図 111 ジェスチャーのチューナー GUI

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 73 で与えられます。

$$\text{Event duration} = No.\,of\,samples\ x\ T_{sample}$$

式 73 ジェスチャーの持続時間

ここで、

No. of Samples = 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。

$$T_{sample} = \frac{1}{\mathrm{Re}fresh\ rate}$$

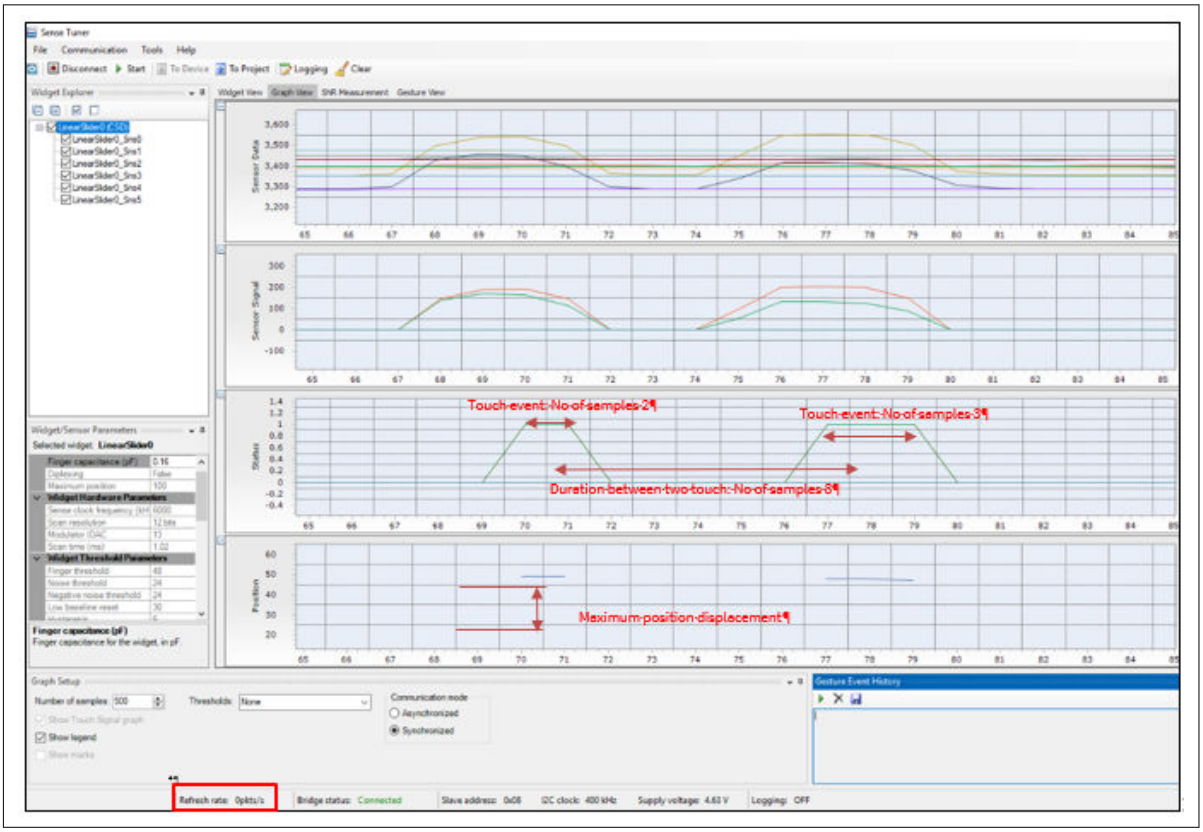


図 112　　チューナー GUI を使用したジェスチャーパラメーターの決定

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。表 26 に、PSOC™ Creator と ModusToolbox™の両方でクリック **ジェスチャー**用に設定する必要の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 ドキュメントを参照してください。表 27 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。

表 26　　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

表 27 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

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。図 113 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。

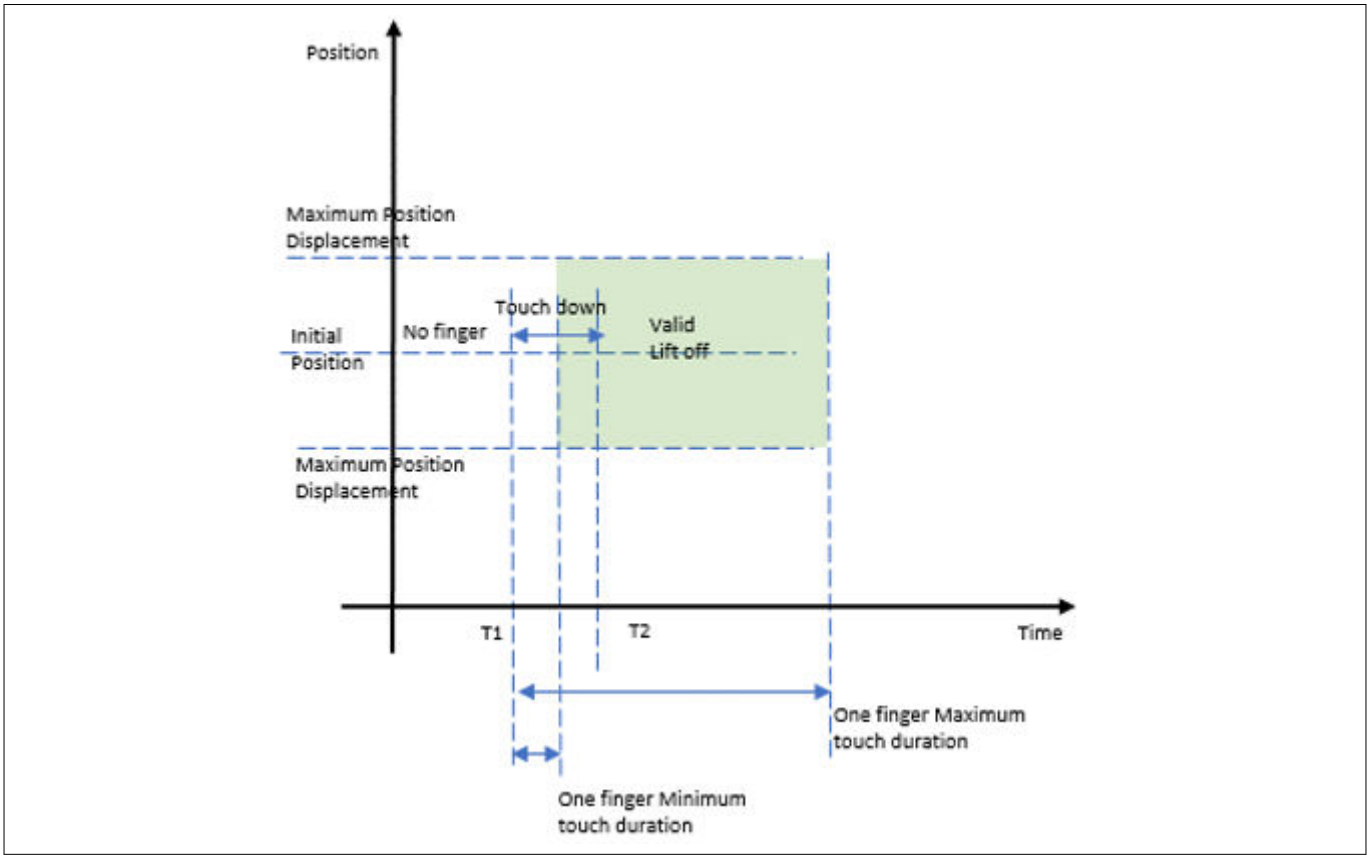


図 113 シングル クリック ジェスチャー

図 113 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、図 112 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 73 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、図 112 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。表 26 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。図 114 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。

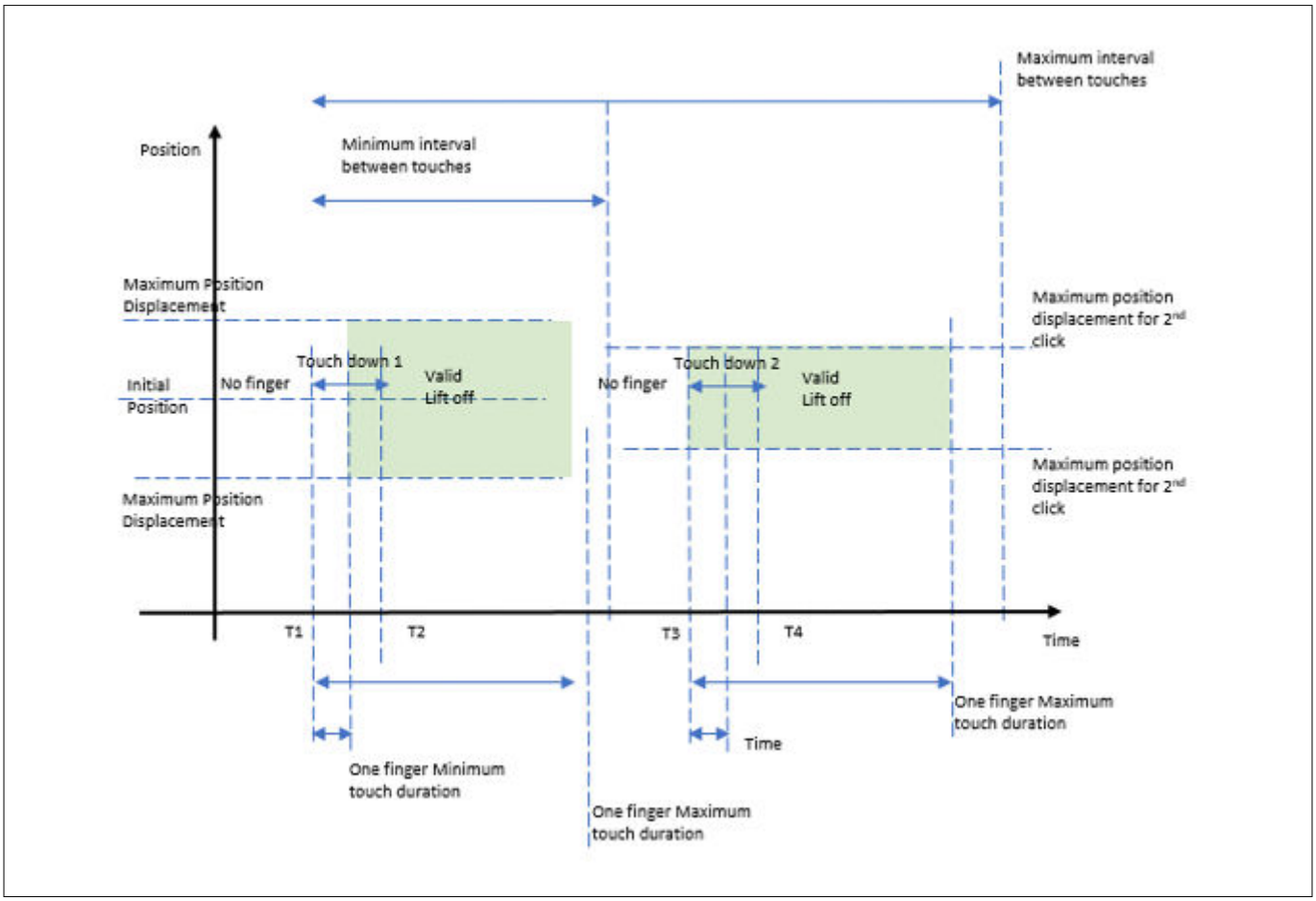


図 114 ダブル クリック ジェスチャー

図 114 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、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図 112 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/ミドルウェア ドキュメントを参照してください。表 28 に、スクロール ジェスチャー用に設定する必要のあるパラメーターを示します。

注: 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ません。

表 28 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	

6.3.4.3.1 1 本指スクロール

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 ドキュメントを参照してください。

表 28 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 73 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 74 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。

$$\text{Position Threshold} = \frac{Change\ in\ Centroid\ position}{Total\ duration\ of\ scroll} \times Duration\ of\ timestamp$$

式 74 位置しきい値を決定する式

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。

   注: スクロールステップパラメーターは ModusToolbox™ では使用できません。

7. 原則として、スライダーの最大位置をスライダーの寸法の 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 (図 111) 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 ドキュメントを参照してください。

ジェスチャーパラメーターを表 28 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 本指スクロールの手順で決定することもできます。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 ドキュメントを参照)。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 本指スクロールと同じ手順に従い、式 73 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。

表 29 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

6.3.5 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 Details** 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 ドキュメントを参照)。図 115 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。

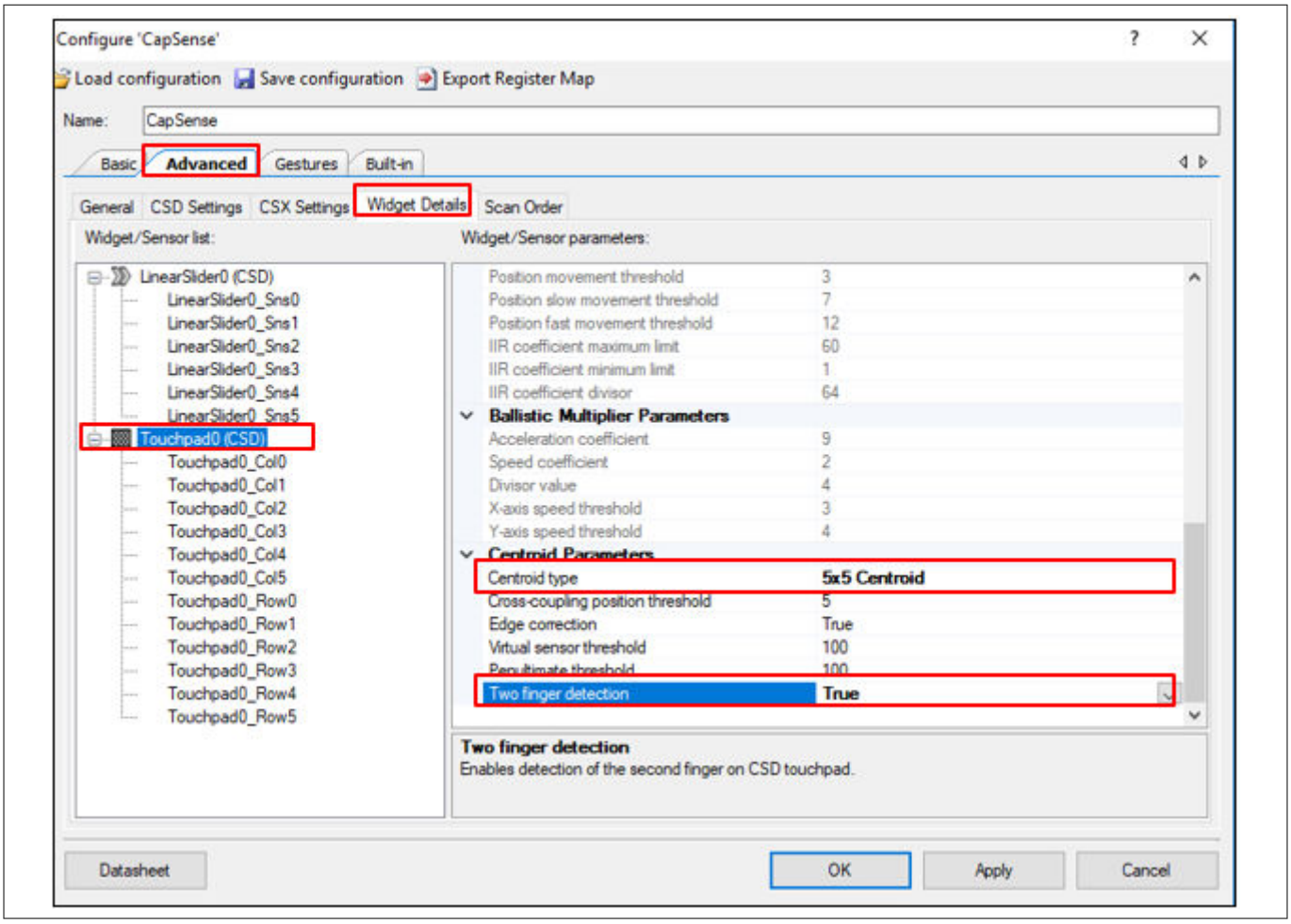


図 115 CAPSENSE™コンポーネントで 2 本指のタッチ ジェスチャーの有効化

6.3.6 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 ドキュメントを参照してください。

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。詳細については、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 を使用できます。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

7 設計上の注意事項

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。

```
/* 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 */
        }
        /* 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 の低消費電力モードに関する基礎知識については、AN234231 - Achieving lowest-power capacitive sensing with PSOC™ 4000T および AN86233 - PSOC™ 4 MCU の低消費電力モードおよび消費電力低減技術を参照してください。PSOC™ 6 MCU 低消費電力モードは AN219528 - PSOC™ 6 MCU 低消費電力モードおよび節電技術、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 低消費電力 CAPSENSE™設計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。

1. 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。
2. 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。
3. 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、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で、他のセンサーをスキャンする必要があります。近接センサーが非アクティブの場合、デバイスはディープスリープモードに戻れます。図 116 に、このプロセスを示します。

7 設計上の注意事項

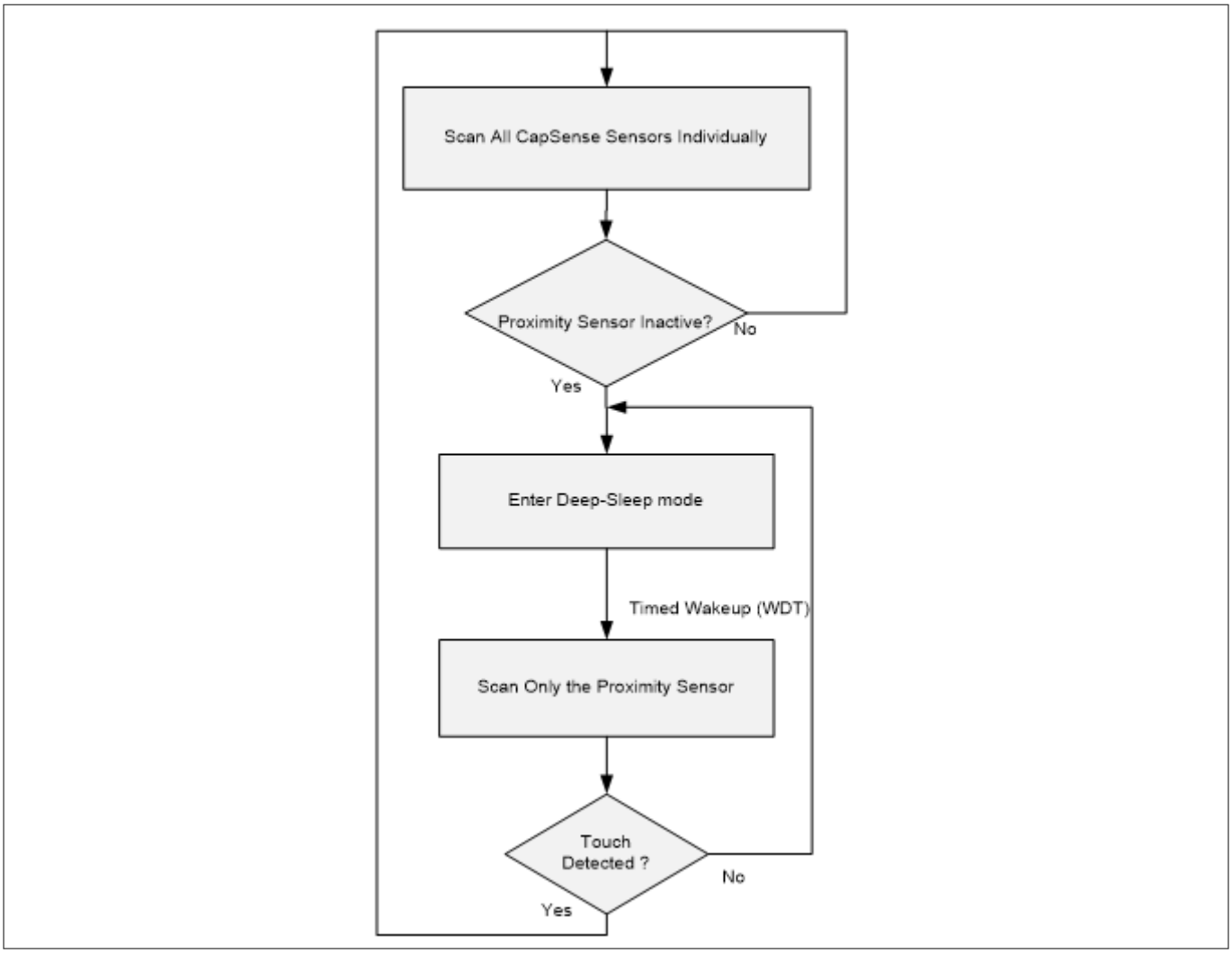


図 116　　低消費電力 CAPSENSE™の設計

1. 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 - CAPSENSE™による近接センシング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 onTouch モードを必要とする CAPSENSE™デザインで役立ちます。手順は、図 116 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。
2. システムタイマーや I^2C 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。
3. 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。

注: 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。

7.1.2 同期スキャン

第 5 世代低消費電力 CAPSENSE™ (MSCLP) では、外部フレームスタート機能を使用してセンサーのスキャンを制御することができます。これにより、外部フレーム開始ピン (ext_frm_start) のパルスによってスキャンをトリガーできます。この機能は PSOC™ 4000T で利用可能です。

この機能は、MSCLP を搭載した複数のデバイスのスキャンを同期させたり、LCD などの外部周期ノイズ源と同期させたりするために使用できます。

外部フレーム開始パルスのガイドライン

外部フレーム開始機能を正しく動作させるための外部フレーム開始信号の特性を以下に示します。

1. **スキャントリガー:** スキャンは、アプリケーション内で scan_all_slots が呼び出された後、外部フレーム開始信号を受信した場合にのみ行われます。

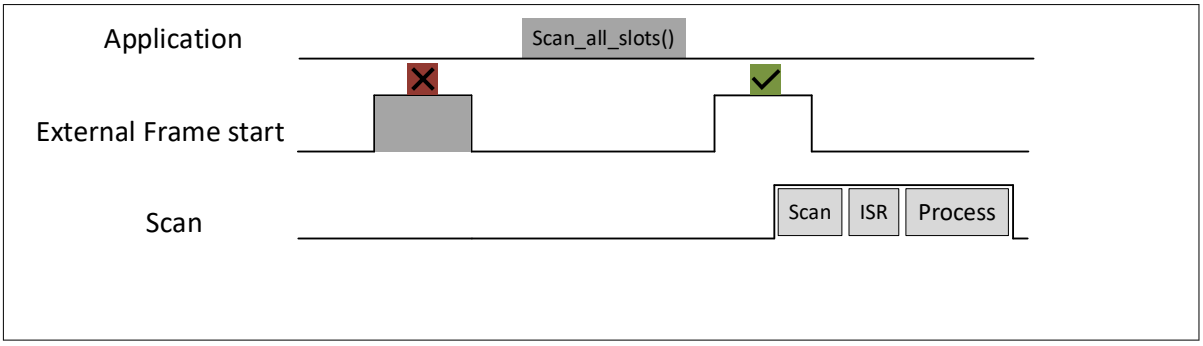


図 117 スキャントリガー

2. **パルスタイプ:** きれいな矩形波
3. **パルス周期:** 後続の 2 つのパルスの間隔であるパルスの周期は、フレーム持続時間 + MSC ISR 持続時間 + 処理持続時間以下にしてはいけません。

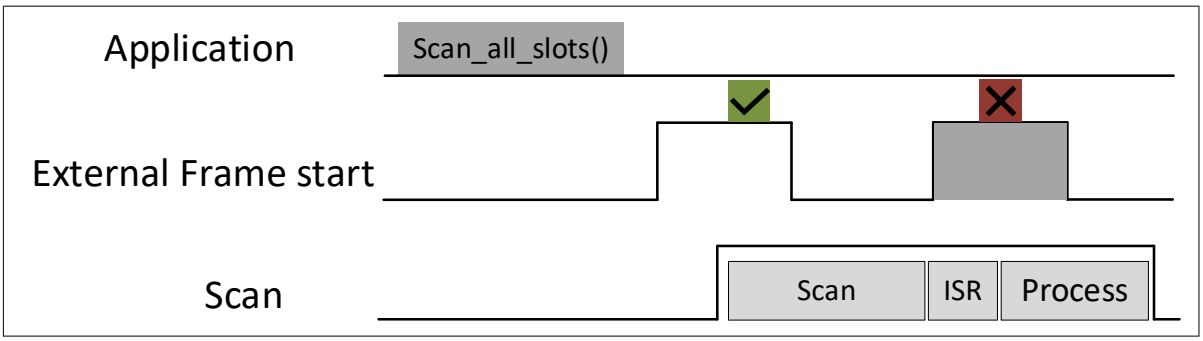


図 118 パルス周期

4. **最小パルス幅:** パルスは 2ILO サイクルより短くしてはいけません。

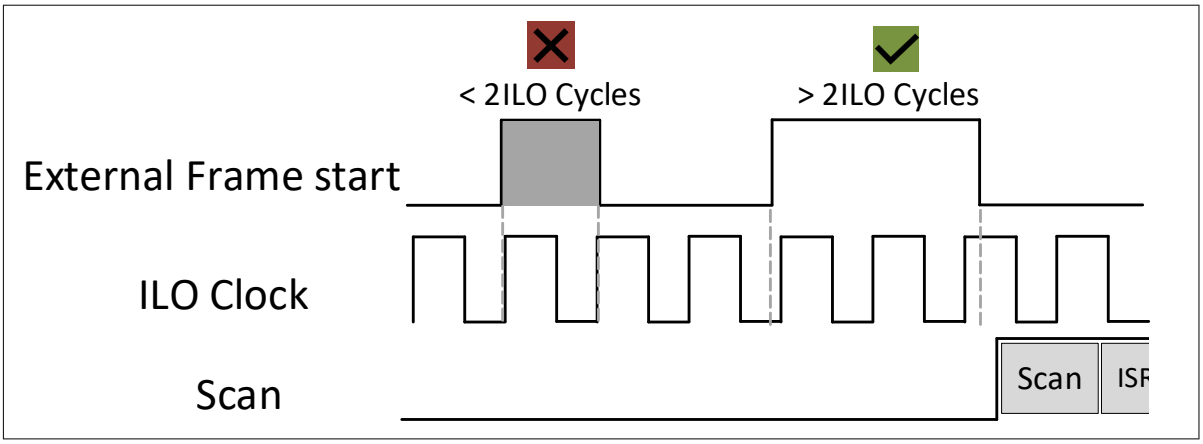


図 119　　最小パルス幅

5. **最大パルス幅:** スキャンフレーム時間より長くしてはいけません。

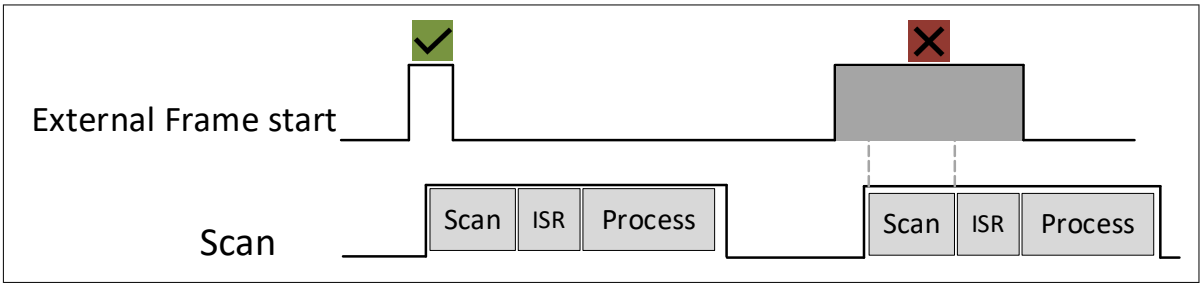


図 120　　最大パルス幅

6. **同期間隔 (scan_all_slots()を呼び出してからパルスが発生するまでの間隔):** ILO のサイクルより長い、あるいは 2 サイクルに相当します。

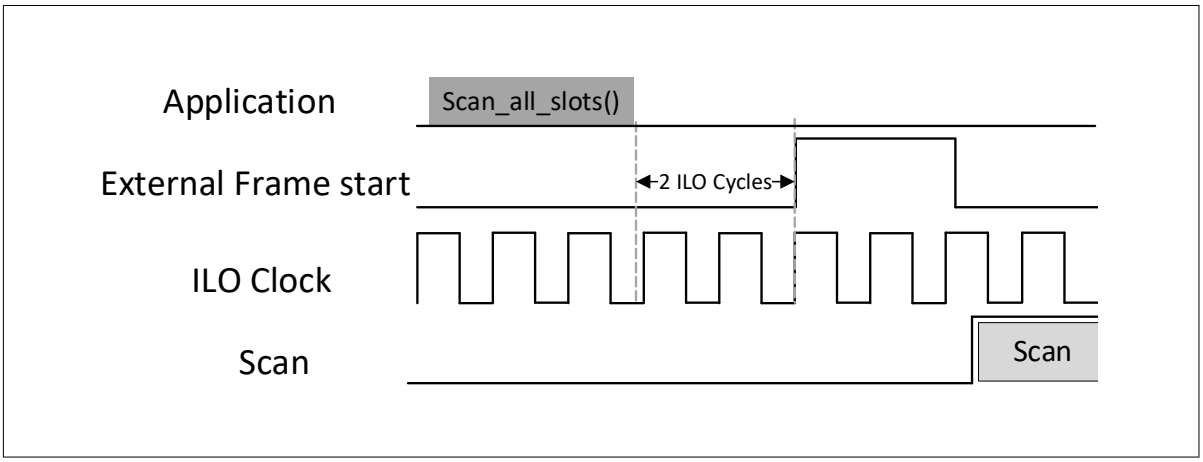


図 121　　同期間隔

外部フレーム開始パルスの調整

MSCLP の外部フレーム開始機能を使用することによって、LCD 同期信号に同期してスキャンできます。外部フレーム開始信号が「**外部フレーム開始パルス ガイドライン**」に記載されているガイドラインに準拠していない場合、信号を調整する必要があります。同期信号を調整する 1 つの方法は、デバイスに存在する TCPWM ハードウェアを使用することです。LCD 同期信号 (例えば 60Hz の矩形波)を受信し、適切な幅の external_frame_start 信号を出力として生成します。

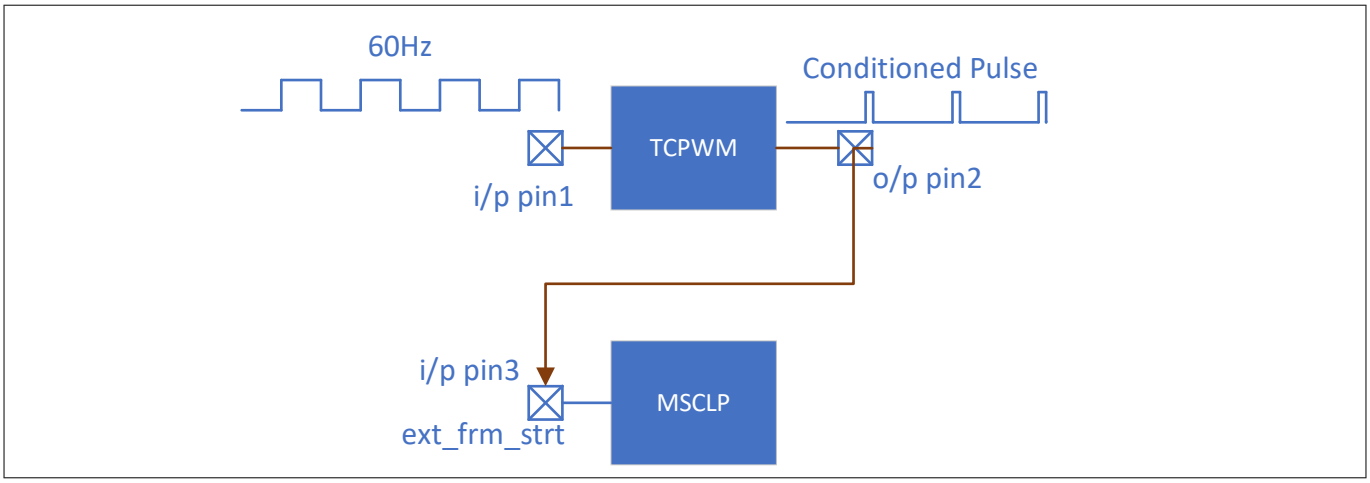


図 122　　TCPWM を使用して条件付きパルスを生成

TCPWM はシングルショットタイマとして構成されます。タイマは外部入力パルスによってトリガされます。この外部入力パルスは LCD 同期信号であり、TCPWM からの出力は MSCLP の ext_frm_start ピンに供給されます。

このアプローチでは、3 つの GPIO と 1 つの TCPWM が使用されます。

1. TCPWM 用外部トリガ入力
2. TCPWM ライン出力
3. MSCLP ext_frm_strt ピン

TCPWM からの出力は、ext_frm_strt ピンの入力に配線して接続する必要があります。

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。

図 123 に、最も一般的な CAPSENSE™センサーの構造を示します。

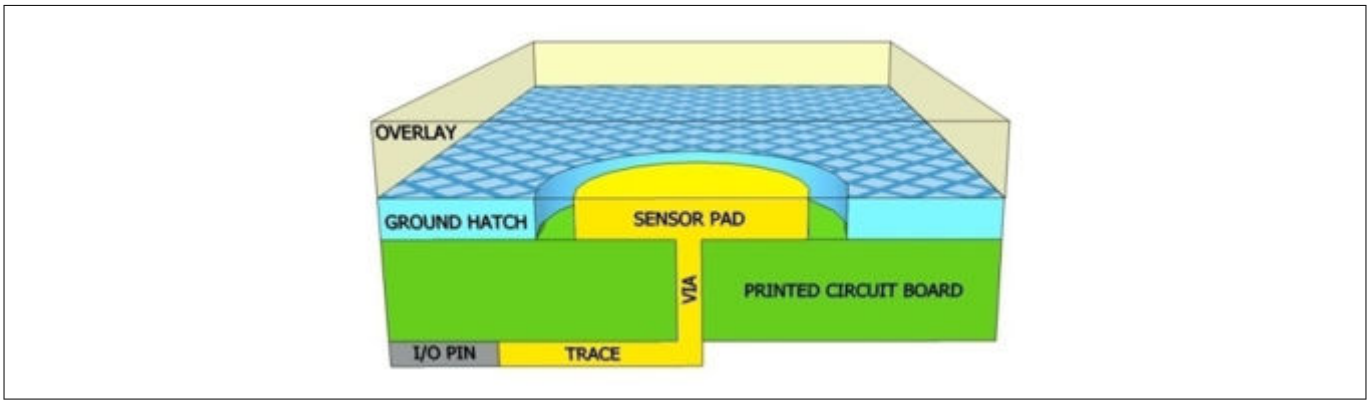


図 123　　CAPSENSE™センサーの構造

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。詳細については、耐液性を参照してください。

7 設計上の注意事項

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。図 124 に CAPSENSE™ハードウェアの分解図を示します。

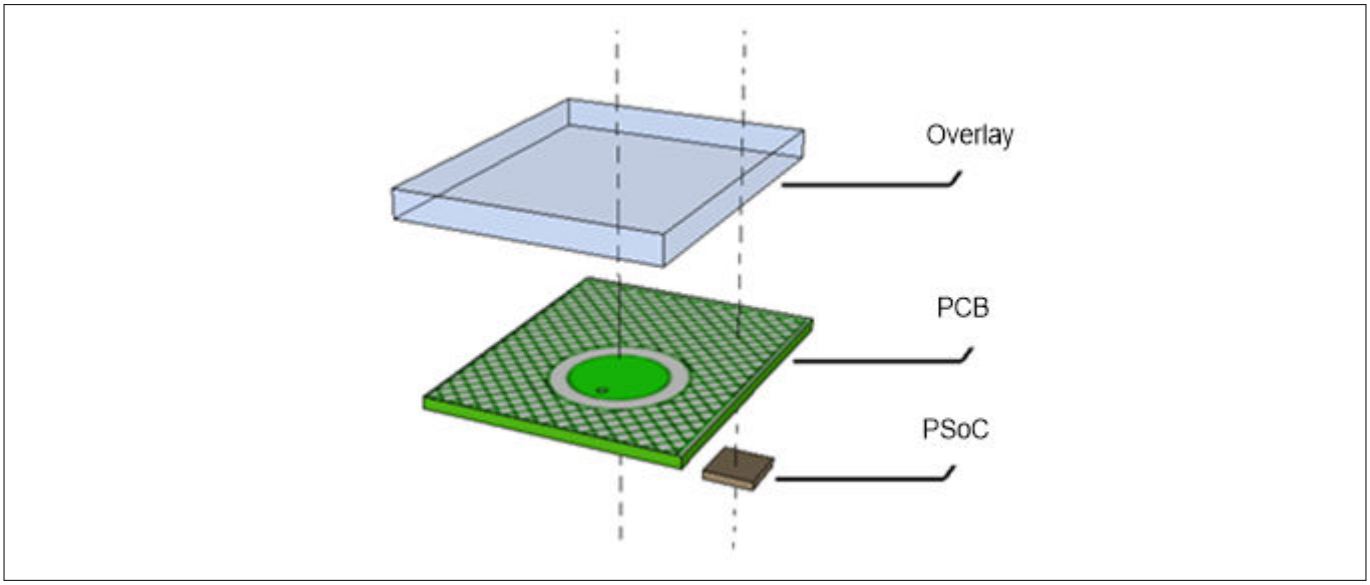


図 124　センサーとしてスプリングを使用する CAPSENSE™ハードウェア センサーの構造

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。図 125 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 CAPSENSE™入門を参照してください。

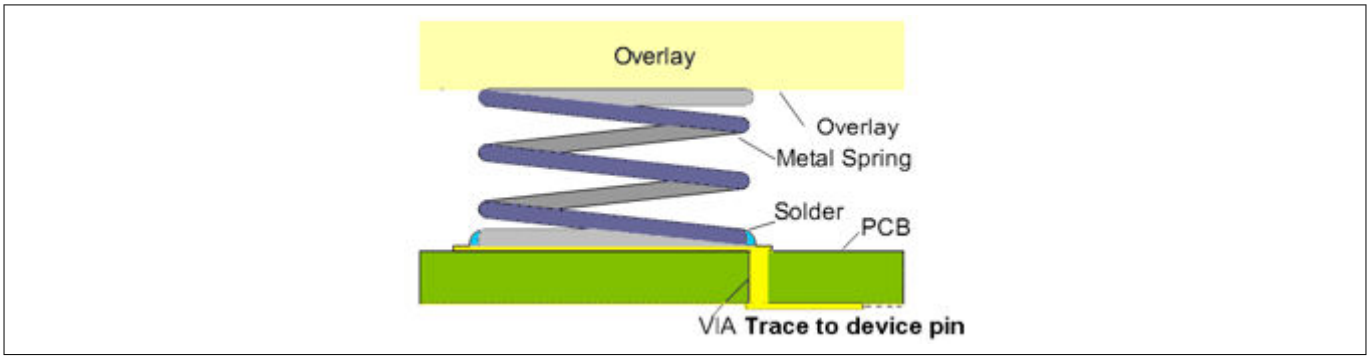


図 125　センサーとしてスプリングを使用するセンサーの構造

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。詳細については、指の静電容量を参照してください。

表 30 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。

表 30　　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
ジプサム (乾式壁)	2.5～6.0

静電容量式タッチ センシングでは、指とセンサーは静電容量の 2 つのプレートとして動作し、オーバーレイが誘電体を形成します。したがって、複数のオーバーレイの場合、2 つのコンデンサーが直列に接続します。誘電率 ε1 と ε2、厚さ d1 と d2 の 2 枚の重ね合わせの実効誘電率を、以下に示します。

$$E_{overall} = \frac{\varepsilon 1 * d1 + \varepsilon 2 * d2}{d1 + d2}$$

式 75　　複数のオーバーレイの実効誘電率

注: *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。詳細については、指の静電容量を参照してください。

表 31 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。

表 31　　アクリル オーバーレイの最大の厚さ

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™
ボタン	5	18
スライダー	5[1)]	18
タッチパッド	0.5	3

1) 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。

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 ($\varepsilon_r = 8$) ですが、アクリルの比誘電率は約 ($\varepsilon_r = 2.5$) です。($\varepsilon_r/2.5$) 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。

1. 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、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 - CAPSENSE™入門を参照してください。
2. 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。
3. スライダーの場合、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。スライダーの設計を参照してください。
4. 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 467 MP と 468 MP 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 (SNR)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 CP

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 ドキュメントにあります。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。C_P は、センサー パッドのサイズとトレースの長さと幅が大きくなると、センサー パッドとグランド 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。

- トレースの長さと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 システムは 85%に校正できない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。

トレースが長いため、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。図 126 に、推奨されるボタン形状を示します。

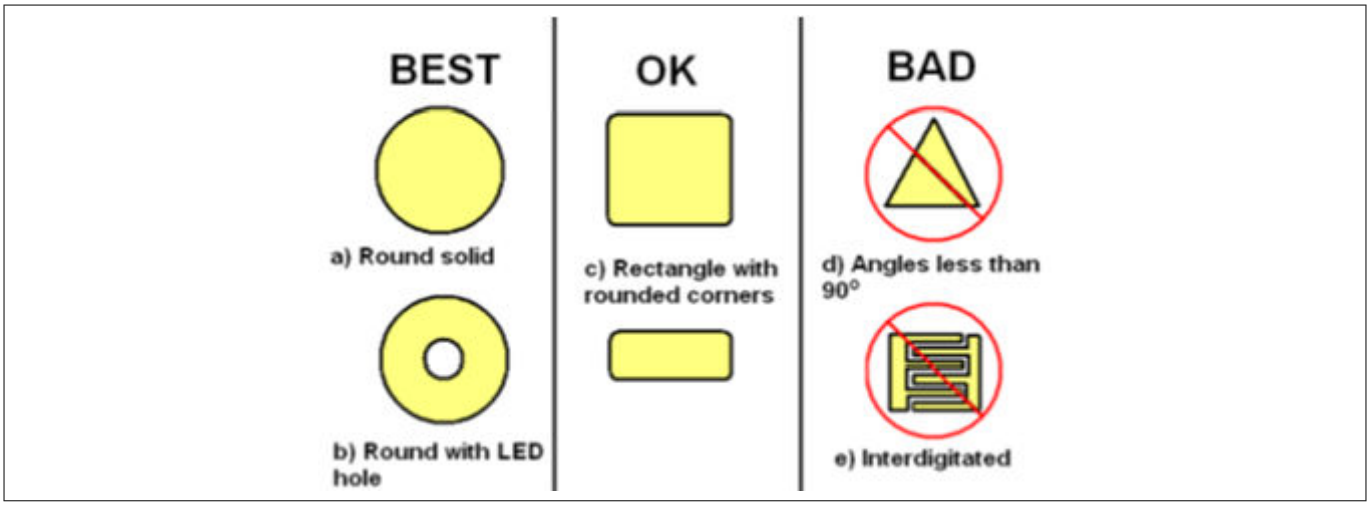


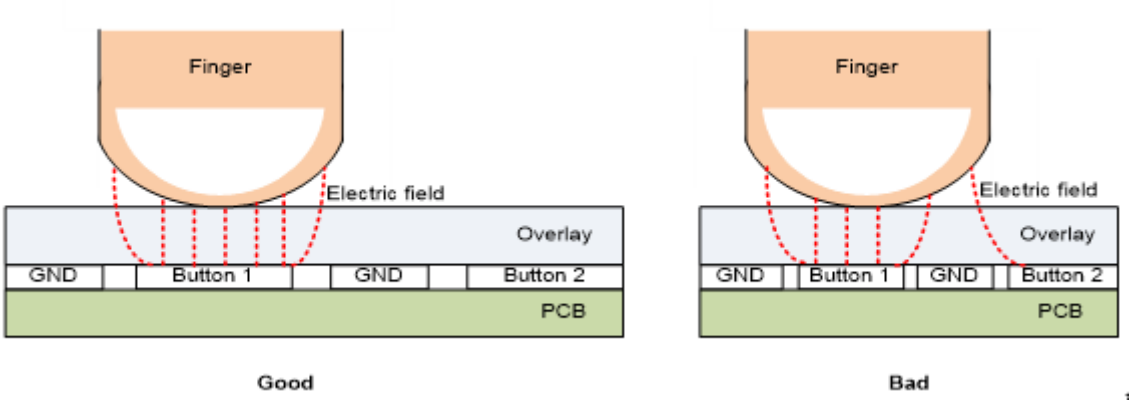
図 126 推奨ボタン形状

サイズ: ボタンの直径は 5mm～15mm の範囲でなければならなりません。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。

図 127 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。

図 127 ボタン間の間隔



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。図 128 に、ハッチングされた地面を備えた上面図と底面図の 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、図 128 に示すようにボイド領域が必要です。

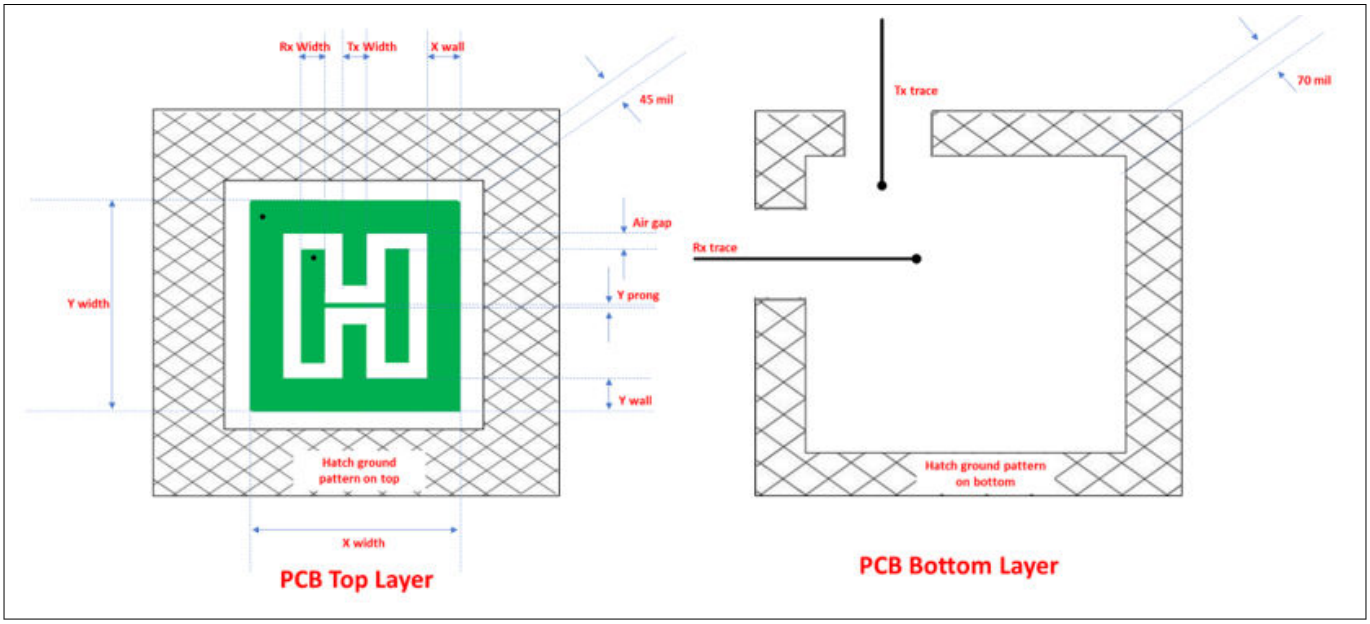


図 128 2 つの Rx プロングを備えた CSX フィッシュボーンボタンパターン

表 32 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。

表 32　　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 プロングの数	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、表 32 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。

***注**:　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 CM 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

図 129 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、表 31 の 13x10mm ボタンを使用し、90°回転させて、図 129 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。

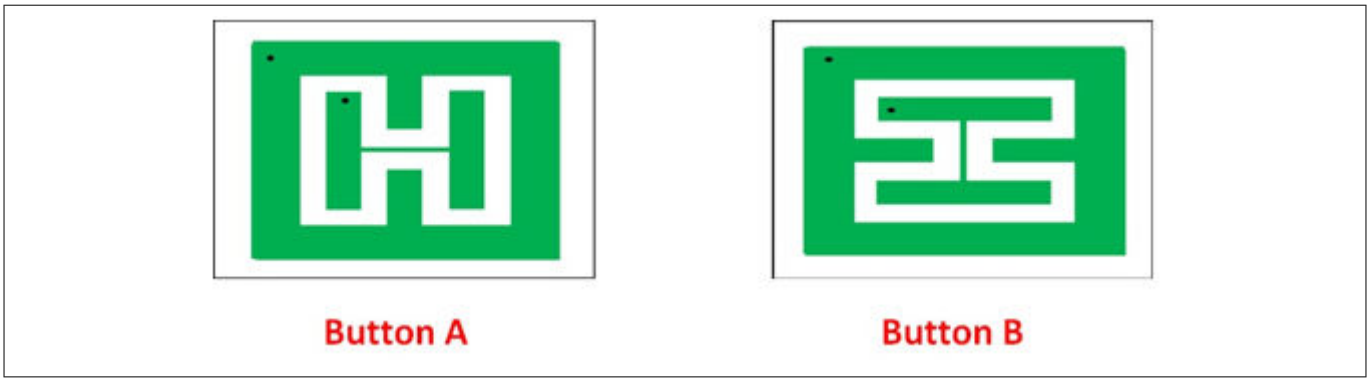


図 129　　Rx プロングの向き

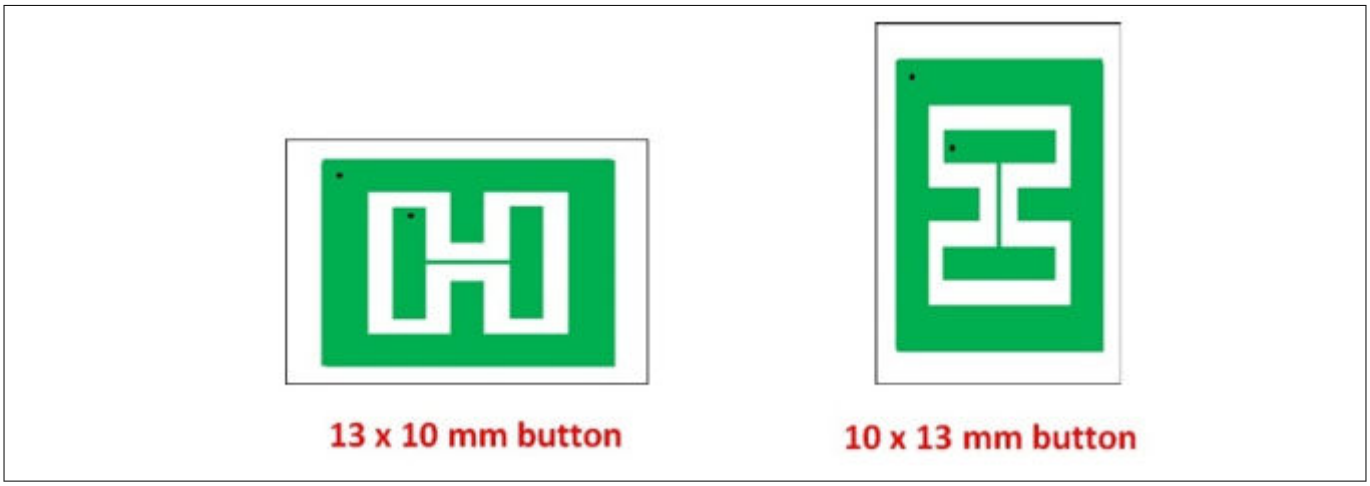


図 130　　ボタンを 90 度回転させて、目的のボタンの寸法を取得

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る場合があります。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、表 31 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、図 131 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。

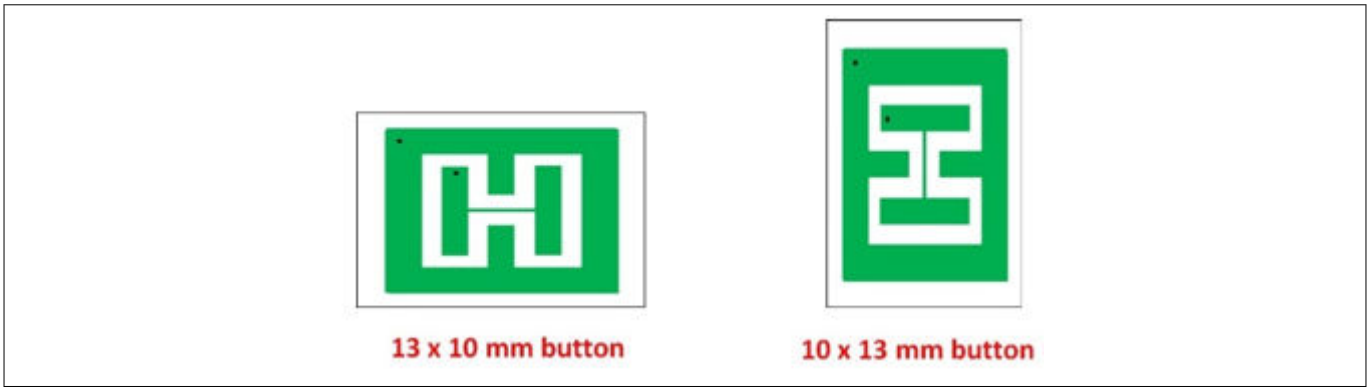


図 131　　任意のパターンに基づく任意の形状のボタンデザイン

表 31 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 のように表 31 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。

$$New_{gap} = TxRx_{gap} \times \left(\frac{ButtonArea_{desired}}{ButtonArea_{reference}} \right)$$

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 New_{gap} 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表 32 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、図 128 を参照してください。

$$Tx_{width} = Rx_{width}$$

$$Tx_{wall} = \frac{Tx_{width}}{2}$$

表 33　　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 プロングの数	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。図 132 に示すように、センサーのサイズを 1 ポイント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。



図 132　　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。ボタン間の間隔を 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の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と、優れたノイズ耐性を取得することに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。

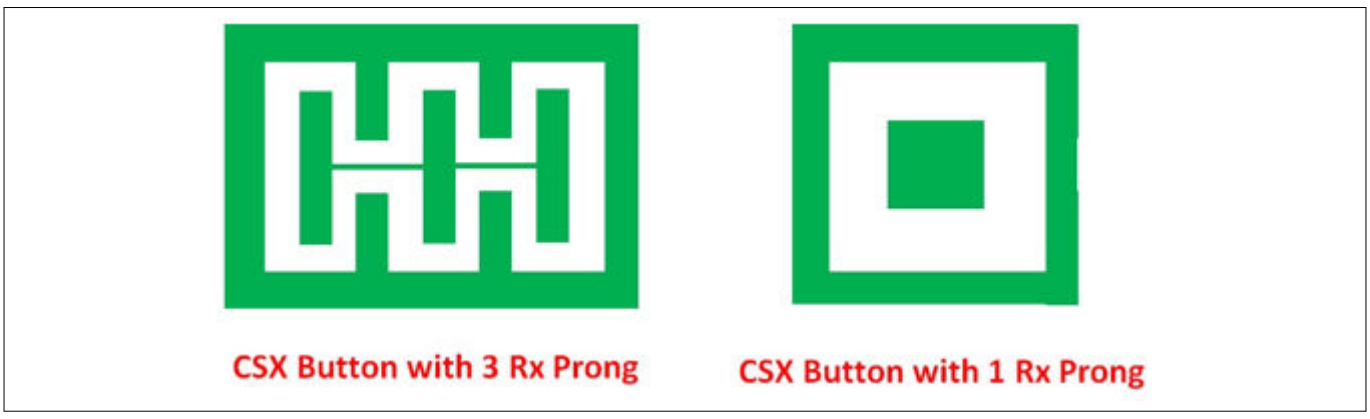
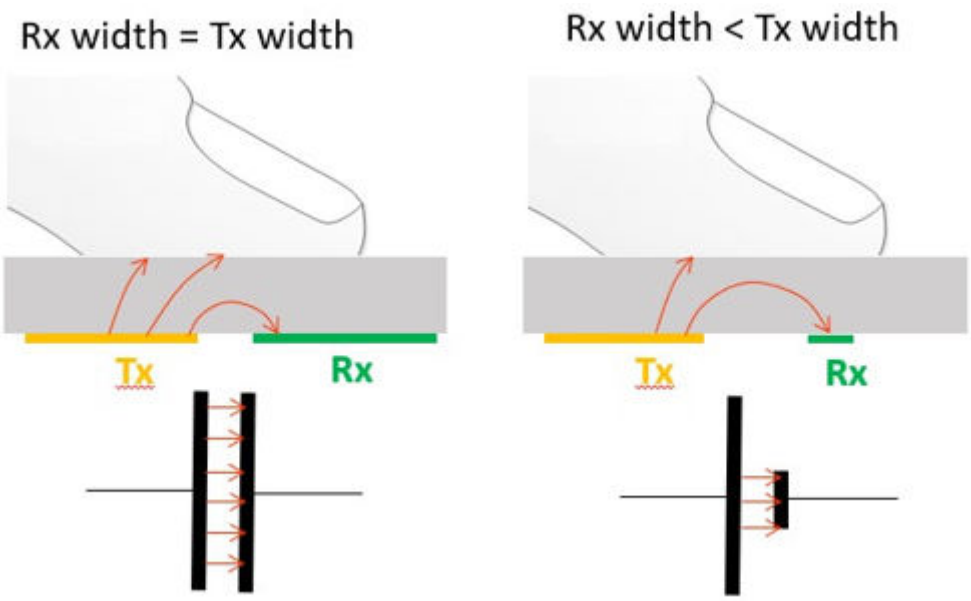


図 133　　プロングの数が異なる相互静電容量ボタン

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。

図 134　　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。

同一平面上のグランド

同一平面上のグランドが存在すると、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ンドプレーンを配置しないことを推奨します。図 128 に、PCB の最上層と最下層の同一平面上のグランドを示します。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。表 33 に、Tx ウォールを増やした 10x10 パターンの例を示します。

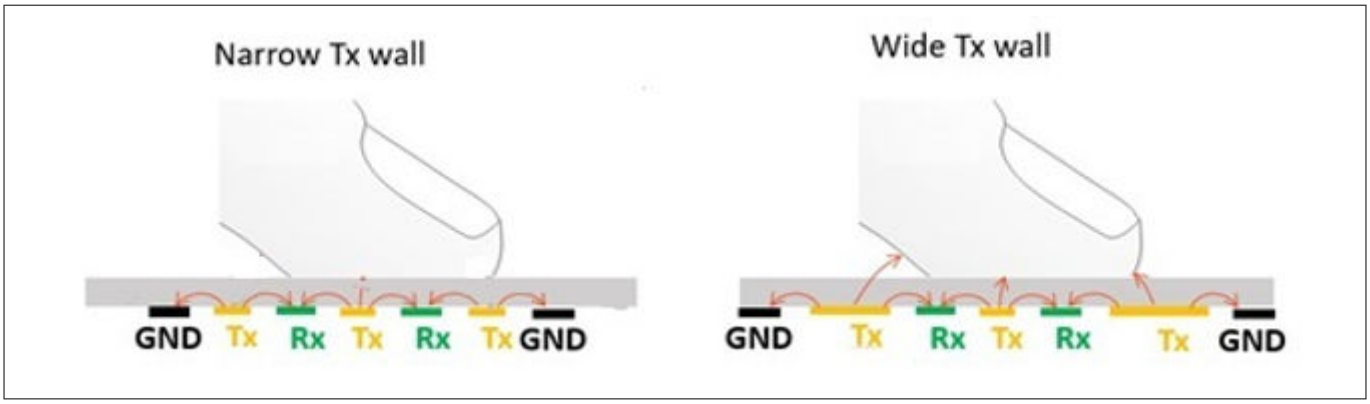


図 135　　Tx 壁幅の影響

表 34　　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

図 136 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、表 35 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については、次の節で説明します。

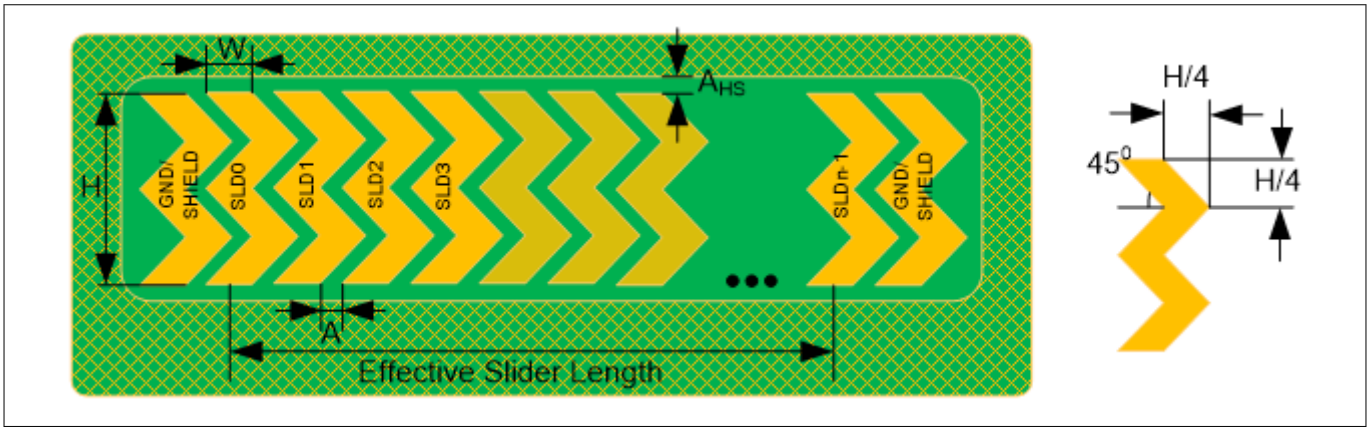


図 136　　標準リニア スライダー パターン

表 35　　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)	1 mm	2 mm	-	8 mm [1)]
	3 mm	4 mm	-	
	4 mm	6 mm	-	
セグメントの高さ (H)	-	7 mm [2)]	15 mm	12 mm
セグメント間のエア ギャップ (A)	-	0.5 mm	2 mm	0.5 mm

(続く)

表 35 (続き) 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
ハッチとスライダー間のエア ギャップ (A_{HS})	-	0.5 mm	2 mm	オーバーレイの厚さに等しい

1) 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。詳細についてはスライダー 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。
2) 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 (SNR) (SNR) ≥ 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。

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

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 SLD0 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 [10]。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。そのため、図 136 に示すように、二重の山形形状を使用することを推奨します。この形状は、図 137 および図 138 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 (図 136 でマークされる、それぞれ寸法「W」と「A」) は、式 48 で述べた関係に従うべきです。

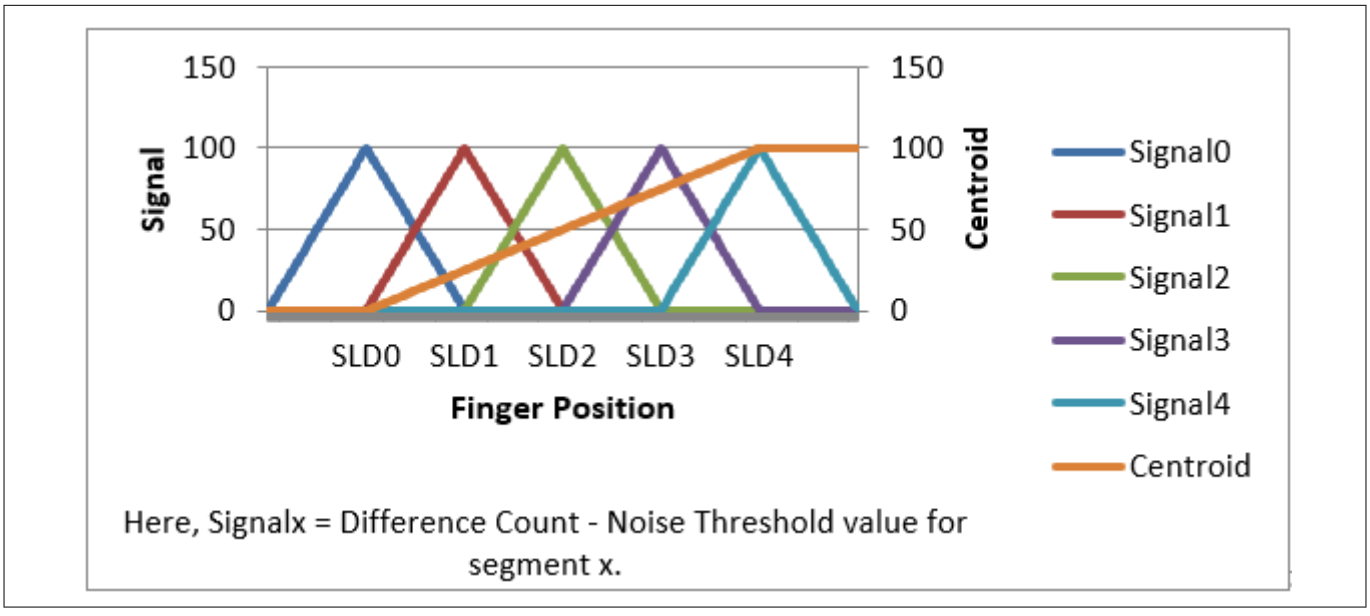


図 137 理想的なスライダーセグメント信号と重心応答

10 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。

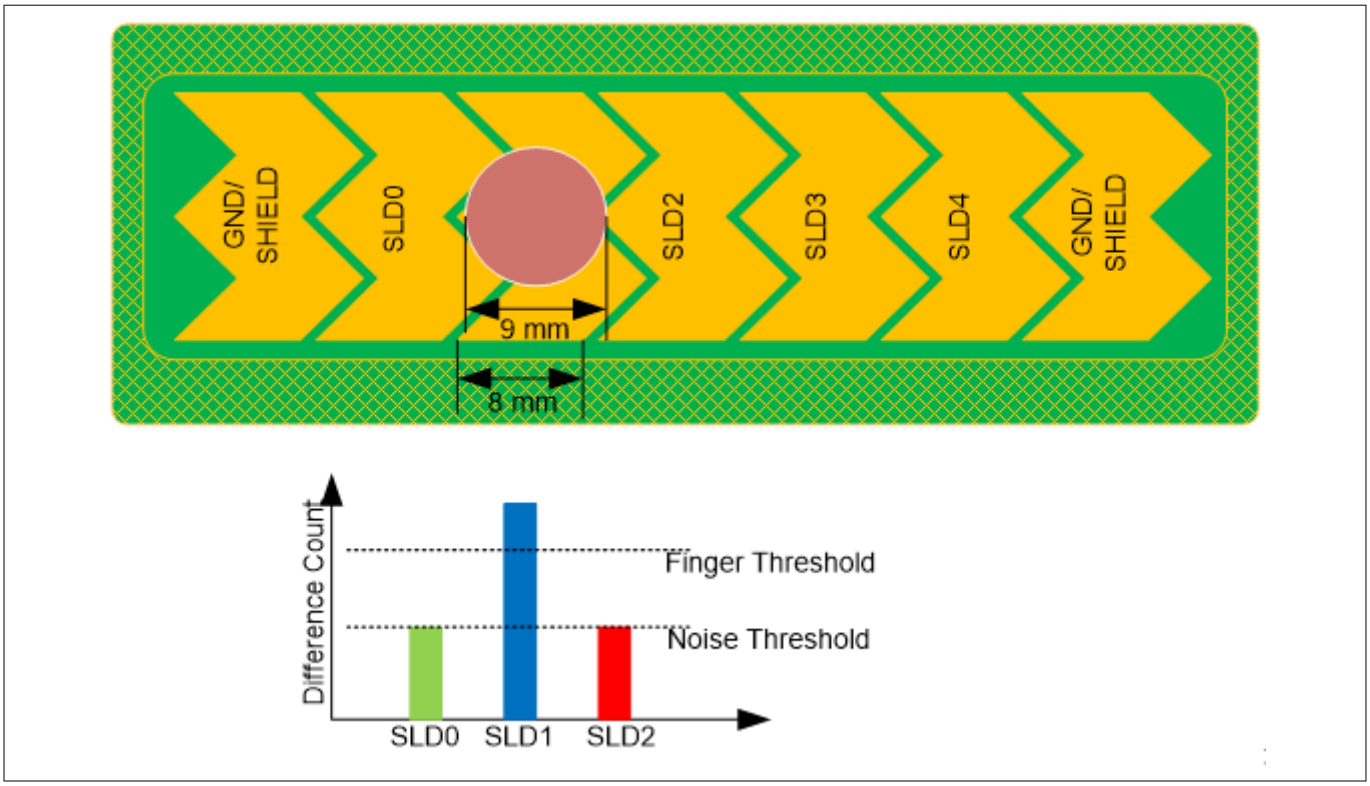


図 138　　理想的なスライダー信号

$$W+2A = finger\ diameter$$

式 76　　セグメント幅、エア ギャップと指の直径の関係

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 76 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 76 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、図 139 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 77 によって計算された重心位置は図 140 のように非線形になります。

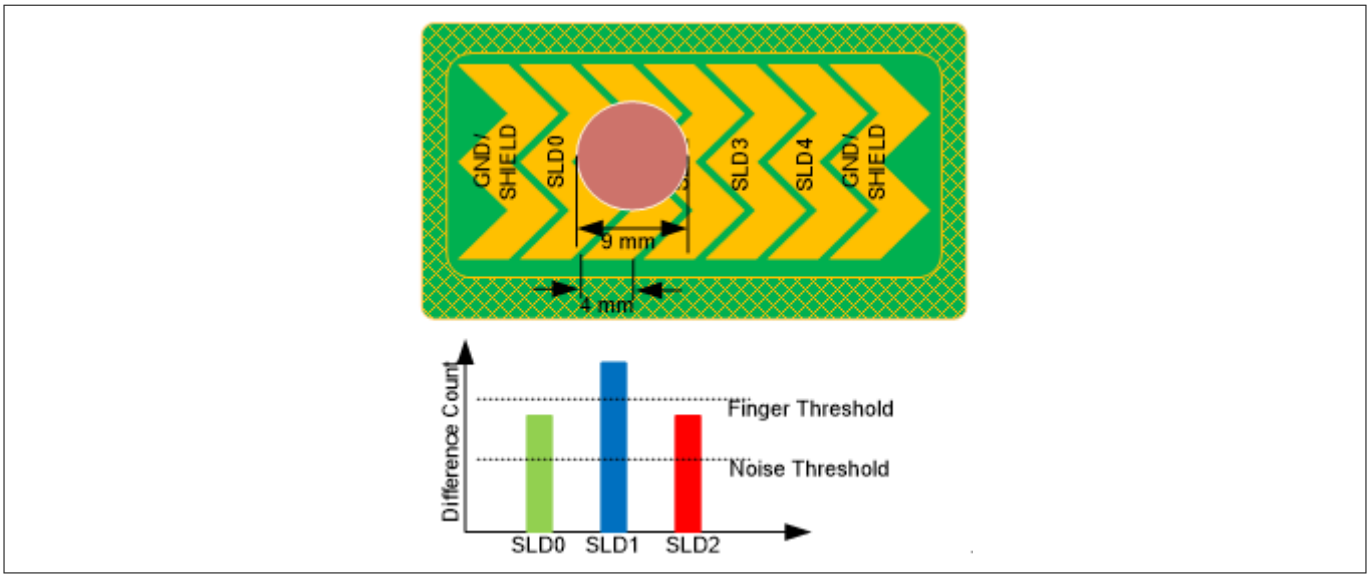


図 139 **スライダーのセグメント幅が推奨値よりも狭いときに、指は 3 つ以上のセグメント上で有効な信号を発生**

$$\text{centroid position} = \left(\frac{S_{x+1} - S_{x-1}}{S_{x+1} + S_x + S_{x-1}} + x\right) \times \frac{\text{Resolution}}{(n-1)}$$

式 77 **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**

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

Si = スライダーセグメントの差分カウント (ノイズ閾値を除く)

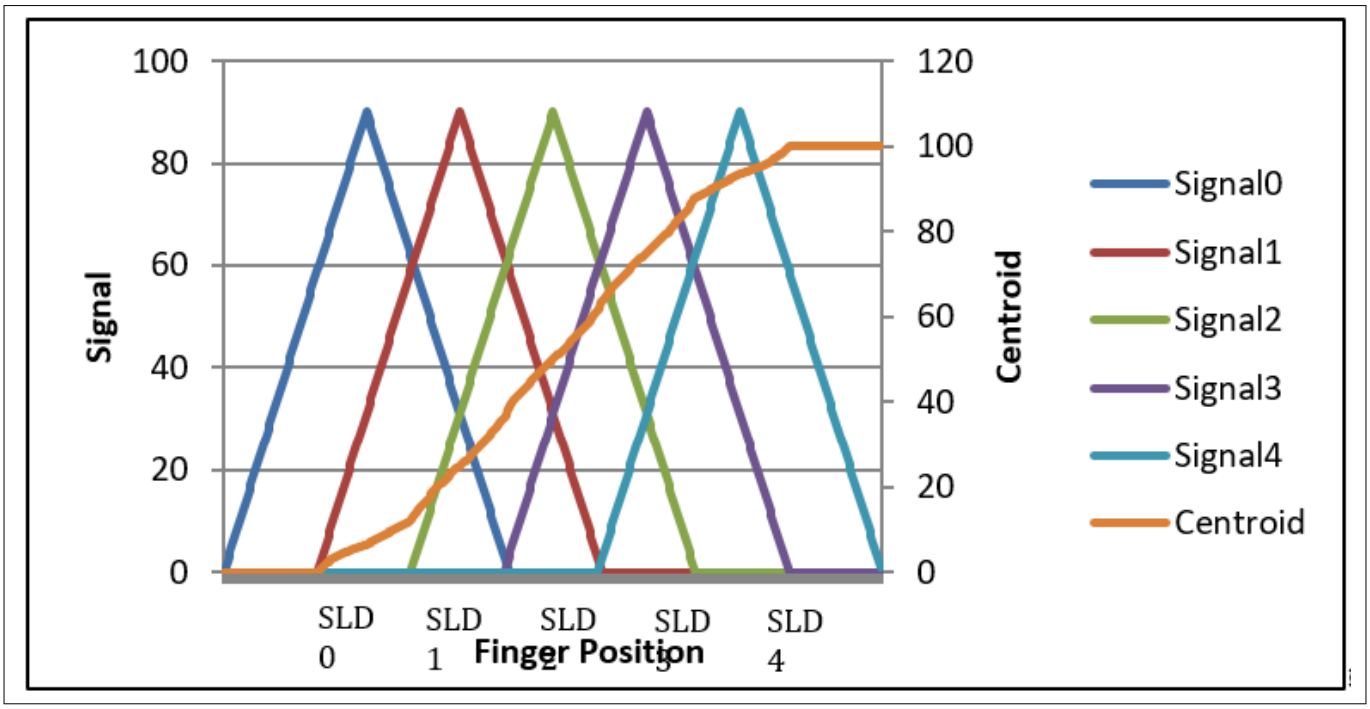


図 140 **スライダー セグメント幅が推奨値より低いときの非線形重心応答**

7 設計上の注意事項

図 140 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 (SNR) ≥ 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、指が静電容量を充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、図 141 に示すように、重心値が PSOC™ Creator では 0xFFFF[11]、ModusToolbox™では 0x0000[12]と報告されます。

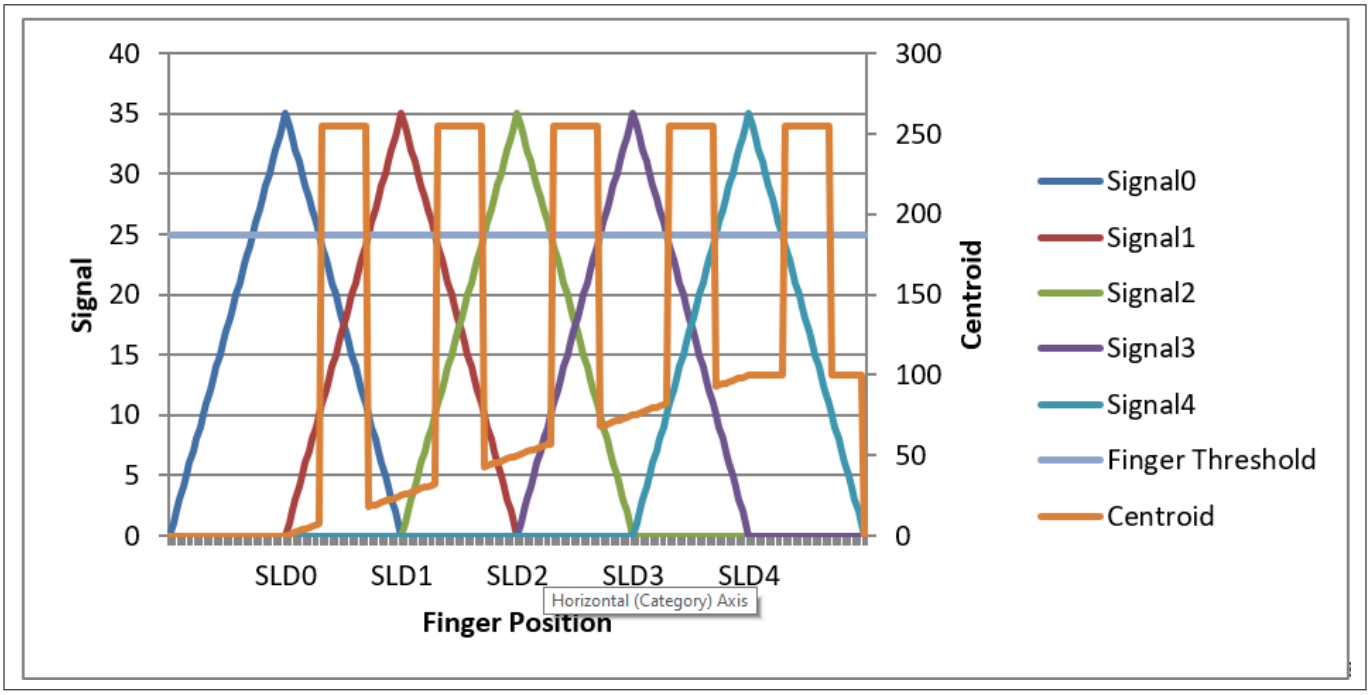


図 141　　スライダー セグメント幅が小さすぎたときに報告された不正な重心

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、表 35 に示します。表 35 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、図 142 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か、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 77 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。表 34 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。

11 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。

12 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。

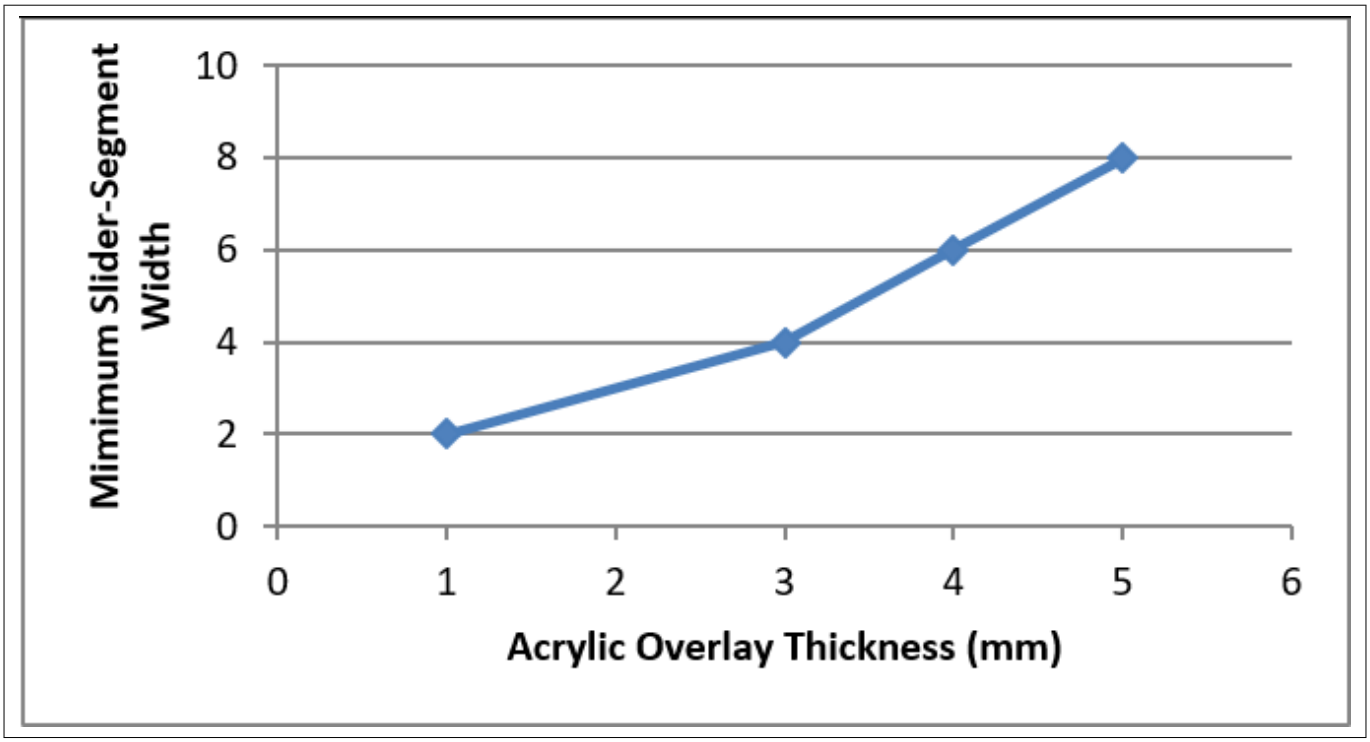


図 142　　アクリル オーバーレイ用のオーバーレイの厚さに対応する最小スライダー セグメント幅

(*slider-segment-width+2*air-gap*) が *指の直径*よりも大きい場合には式 76 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、図 143 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、図 144 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。

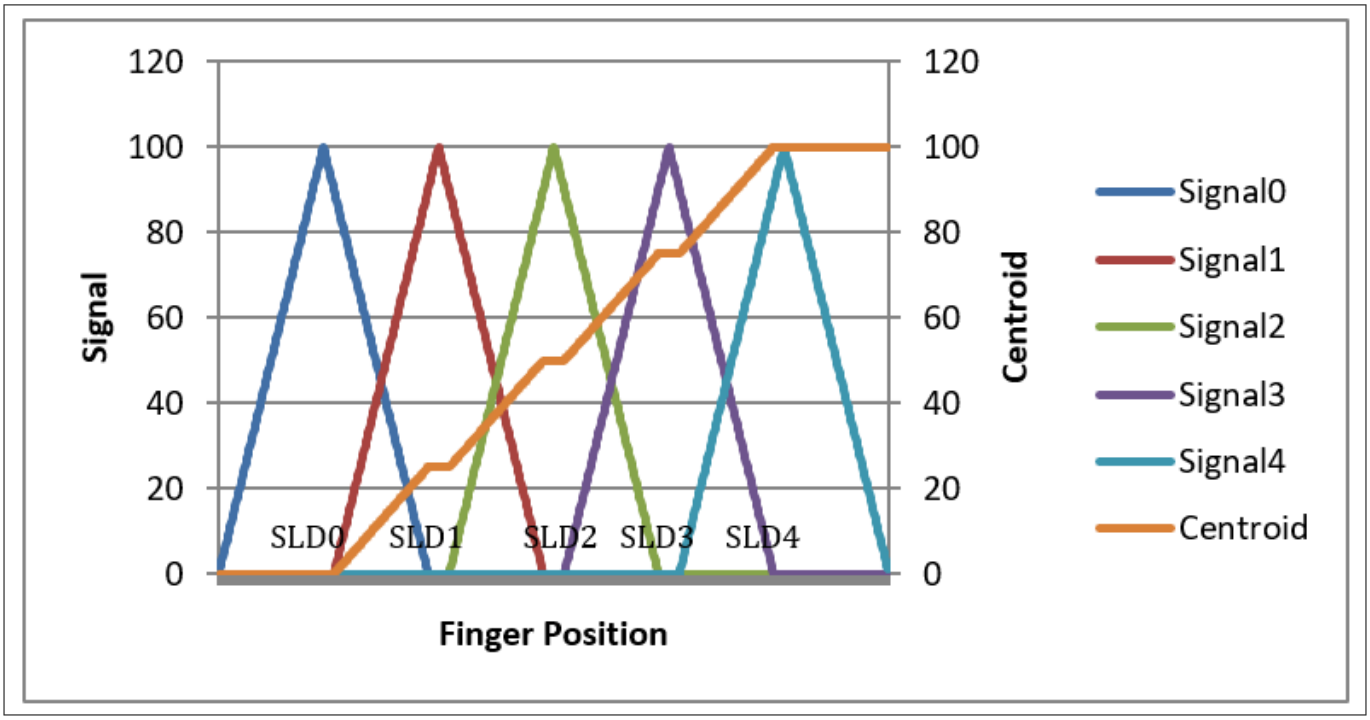


図 143　　スライダー セグメント幅が推奨値より大きい場合のフラット スポット (非応答重心)

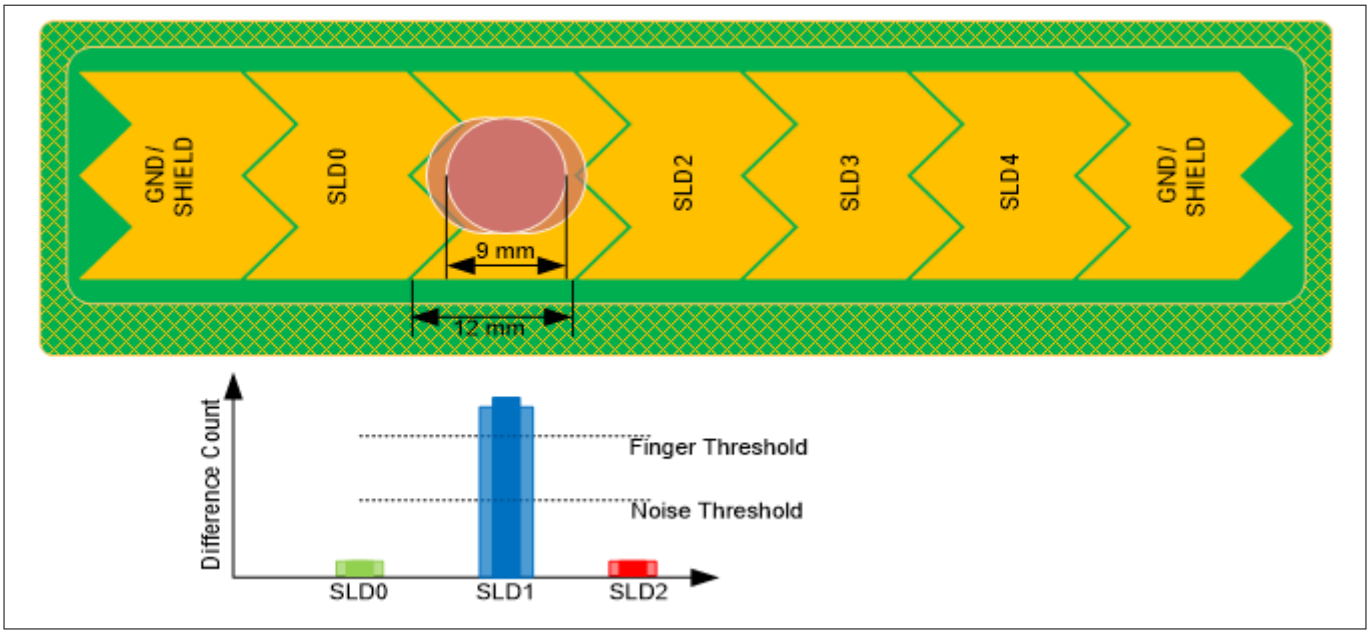


図 144　　スライダー セグメント幅が推奨値より大きい場合のスライダー セグメント上の信号

(*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ことに注意してください (図 137 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、図 136 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、図 145 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。詳細について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。



図 145　　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

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。

1. 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。これは図 136 に示す「有効なスライダー長」と同じです。
2. 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、使用する高さは表 35 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。
3. 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表 35 に示す推奨値にしなければなりません。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。
4. 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。

$$\text{Number of segments} = \frac{slider\ length}{slider\ segment\ width\ +\ \ airgap} + 1$$

式 78　　スライダーに必要なセグメント数

***注**:　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す。*

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、図 143 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。

***注**:　図 136 に示すように、PCB は必要なスライダー長より大きい長さです。PCB の長さは、式 79 に示すようにスライダーの長さに関係します。*

$$\text{PCB length} = \ Slider\ length\ + 3\ \times slider\ segment\ width\ + 2 \times air\ gap$$

式 79　　最小 PCB 長さとスライダー長の関係

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 80 に示すようになります。

$$\text{PCB length} = \textit{Slider length} + \textit{Slider segment width}$$

式 80　　最小 PCB 長さとスライダー長の関係

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。

1. 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。
2. スライダーセグメントの 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、各スライダーセグメントの 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ーセグメントの最大 C_P 44%です。C_P の変動がこの制限を超えると、スライダーセグメント間の感度が変化し、スライダー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ー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と幅は、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 付きスライダー 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。図 146 に示すように、スライダー 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ー 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き、スライダー 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、スライダー 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 以上を達成するために、必要な最小スライダー幅については表 35 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。

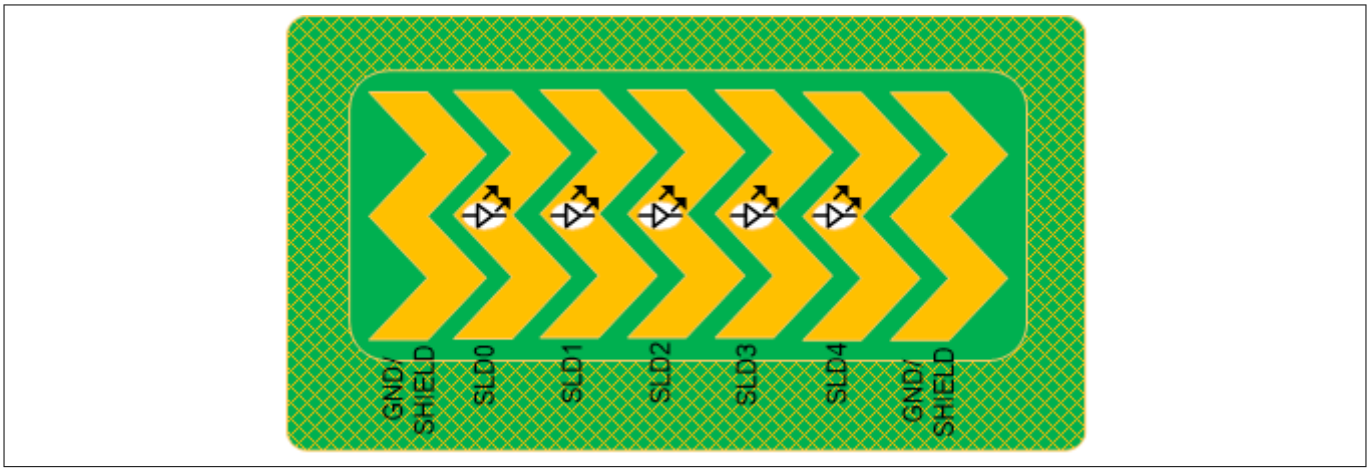


図 146　　LED バックライト付きスライダー設計

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。詳細については 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。
- 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) あけてください。

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、交差部が図 147 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。

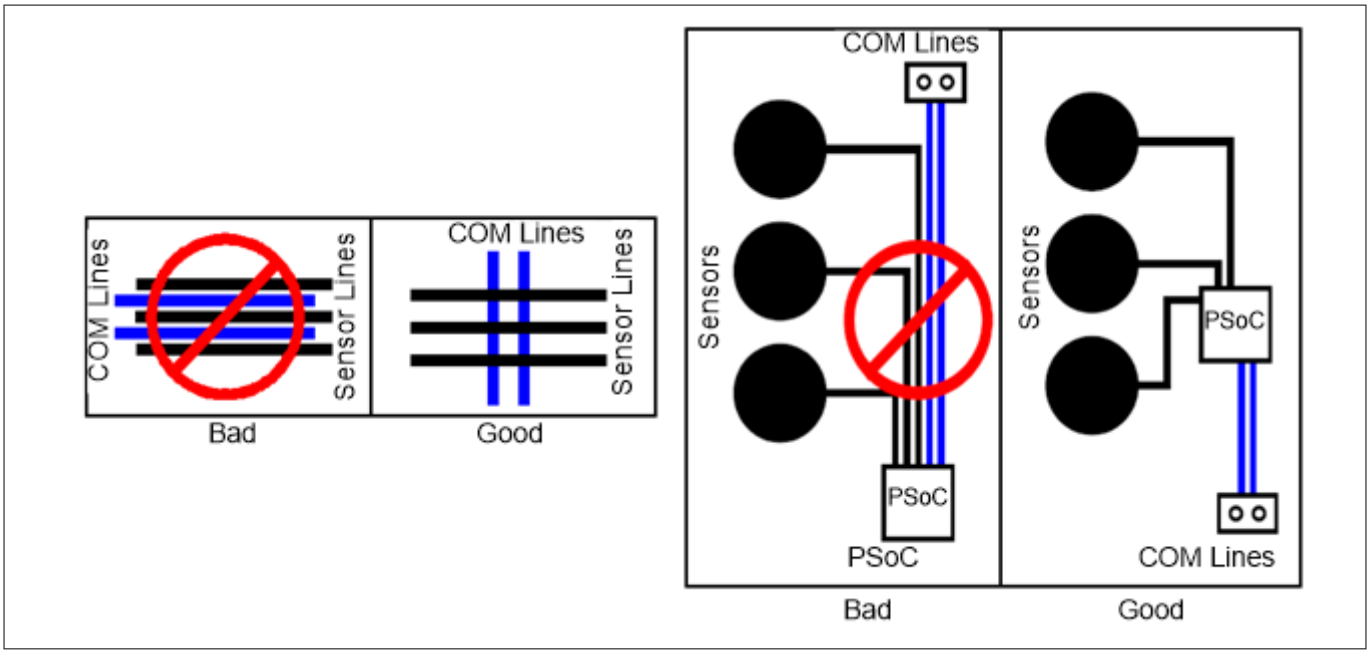


図 147 センサーと通信ラインの配線経路

7 設計上の注意事項

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、図 148 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。

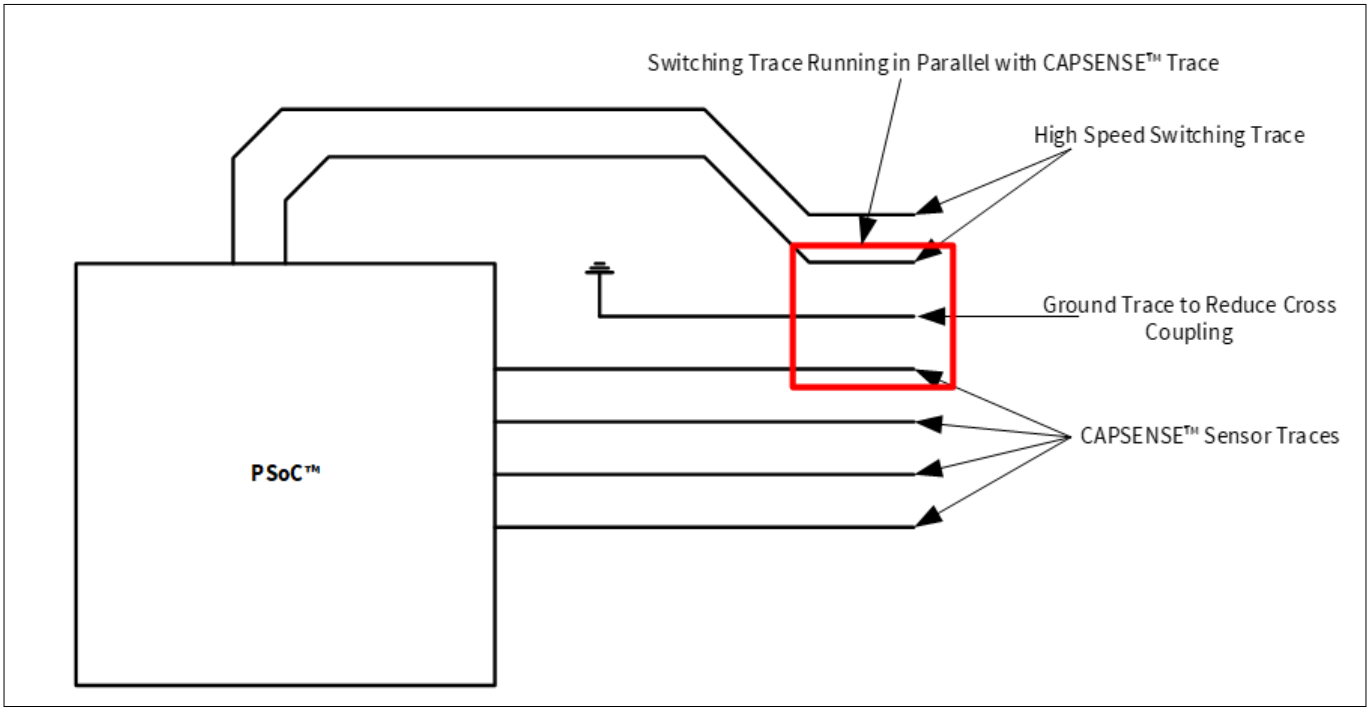


図 148　　高速のスイッチング配線と CAPSENSE™配線間のクロストークを削減

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、図 149 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。

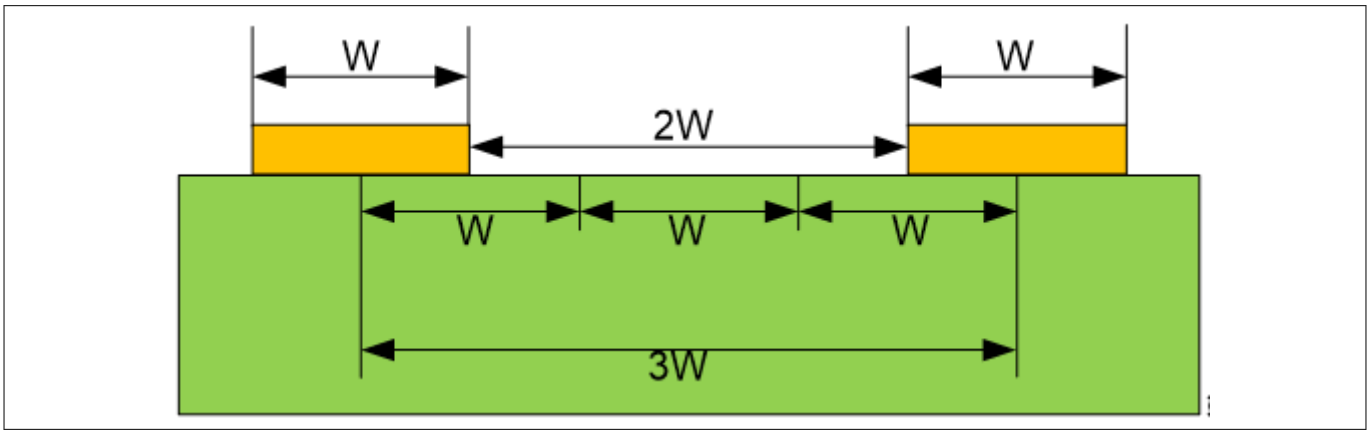


図 149　　クロストークを最低限するために 3W ルールに従う配線間隔

- 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。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、図 150 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。

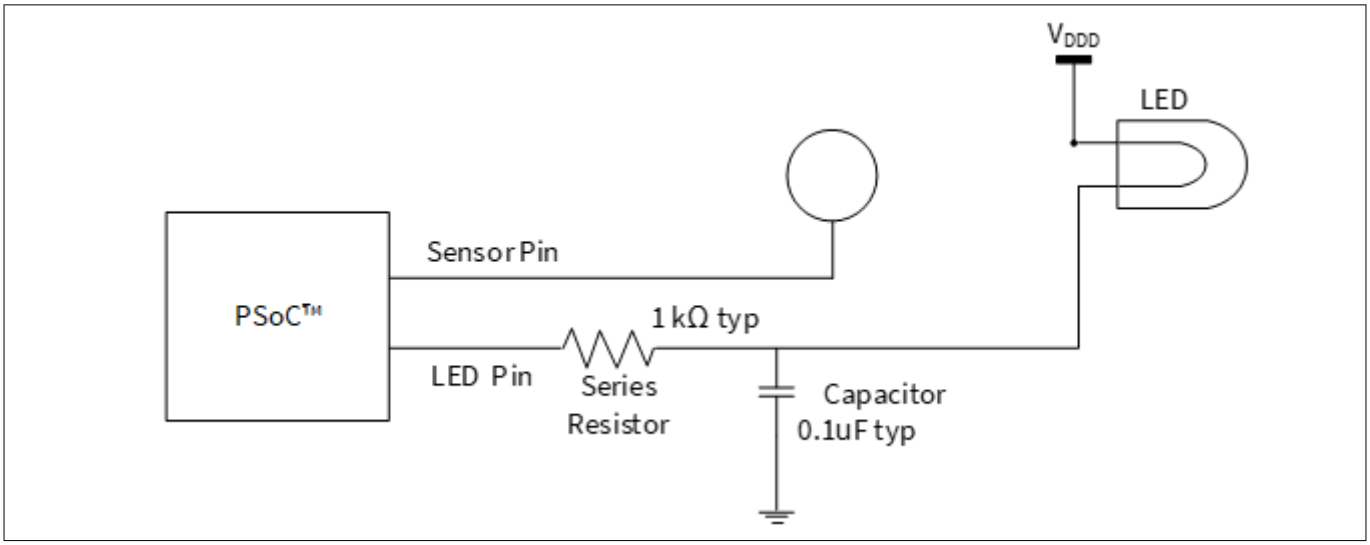


図 150 クロストーク削減

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。図 151 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。

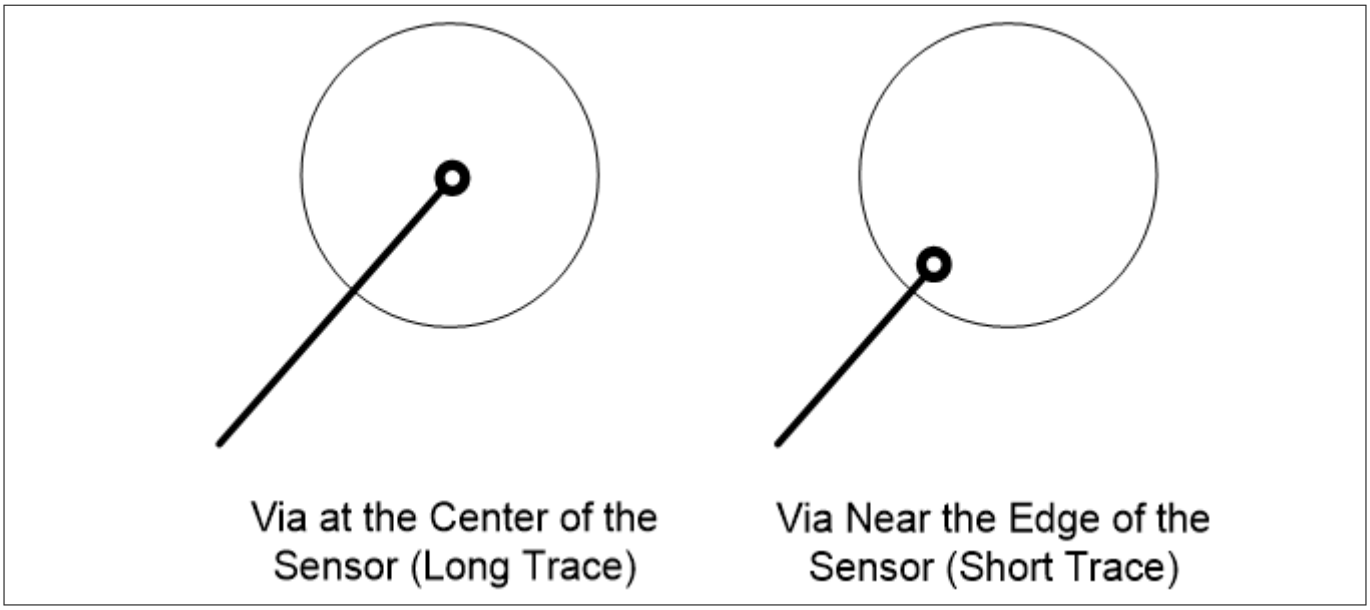


図 151　センサー パッド上のビアの配置

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を従ってください。

- 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。
- 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、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。

図 152 にスター接続を示します。各ファミリの中心点は表 36 に示します。

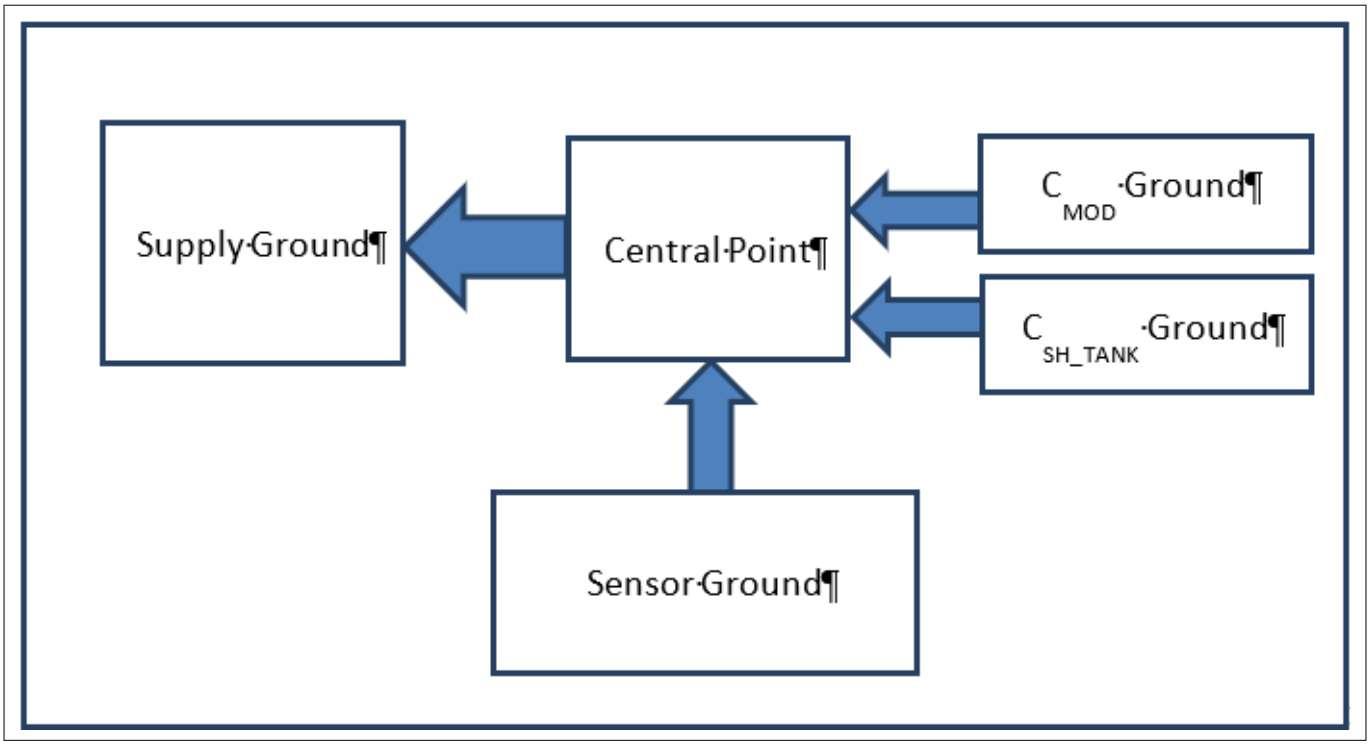


図 152　　グランド面のスター接続

表 36　　スター接続の中心点

ファミリー	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。

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図 153 および図 154 に示します。

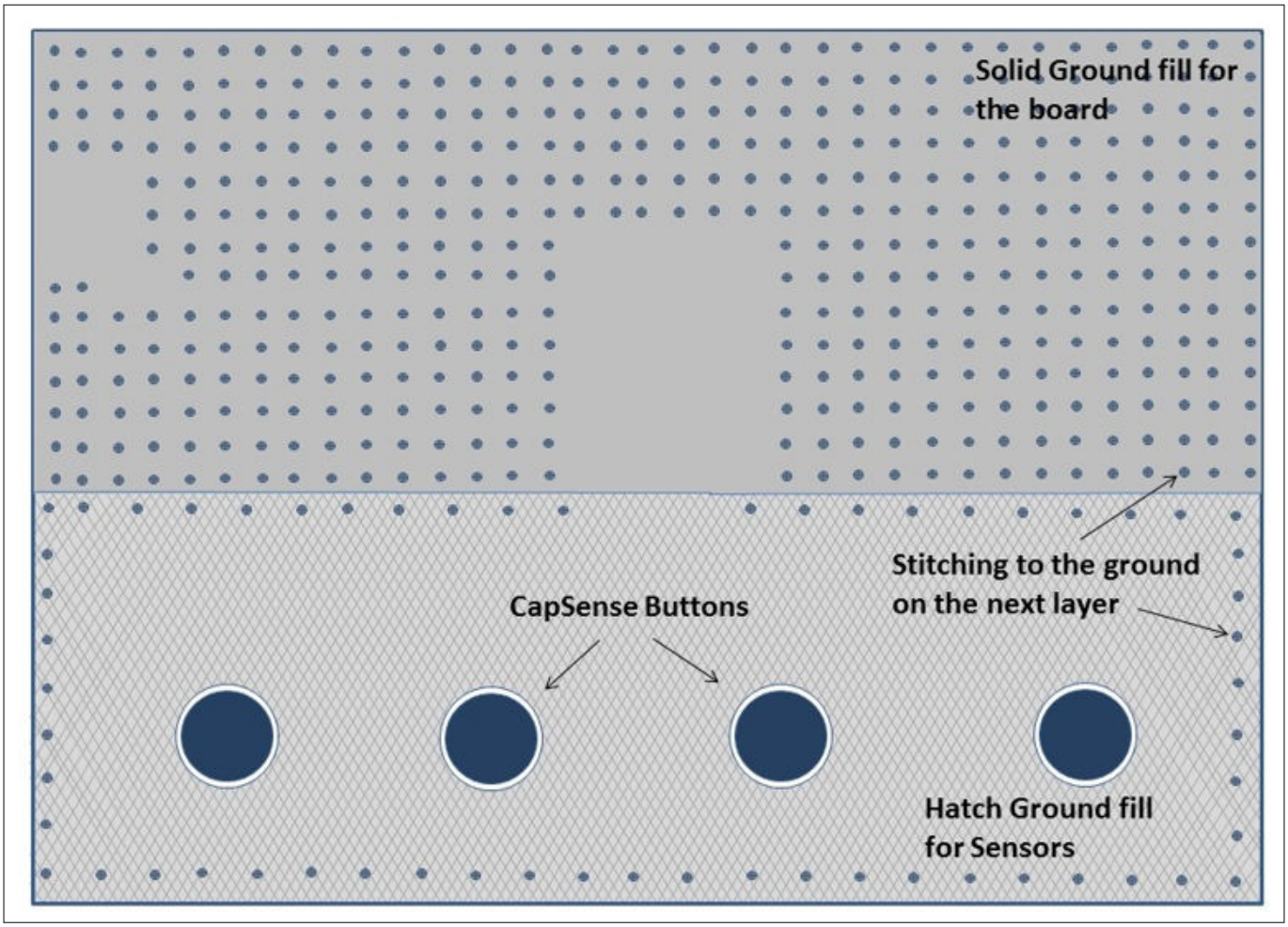


図 153　　E-pad がないチップを使用する場合の PCB 最上層のレイアウト

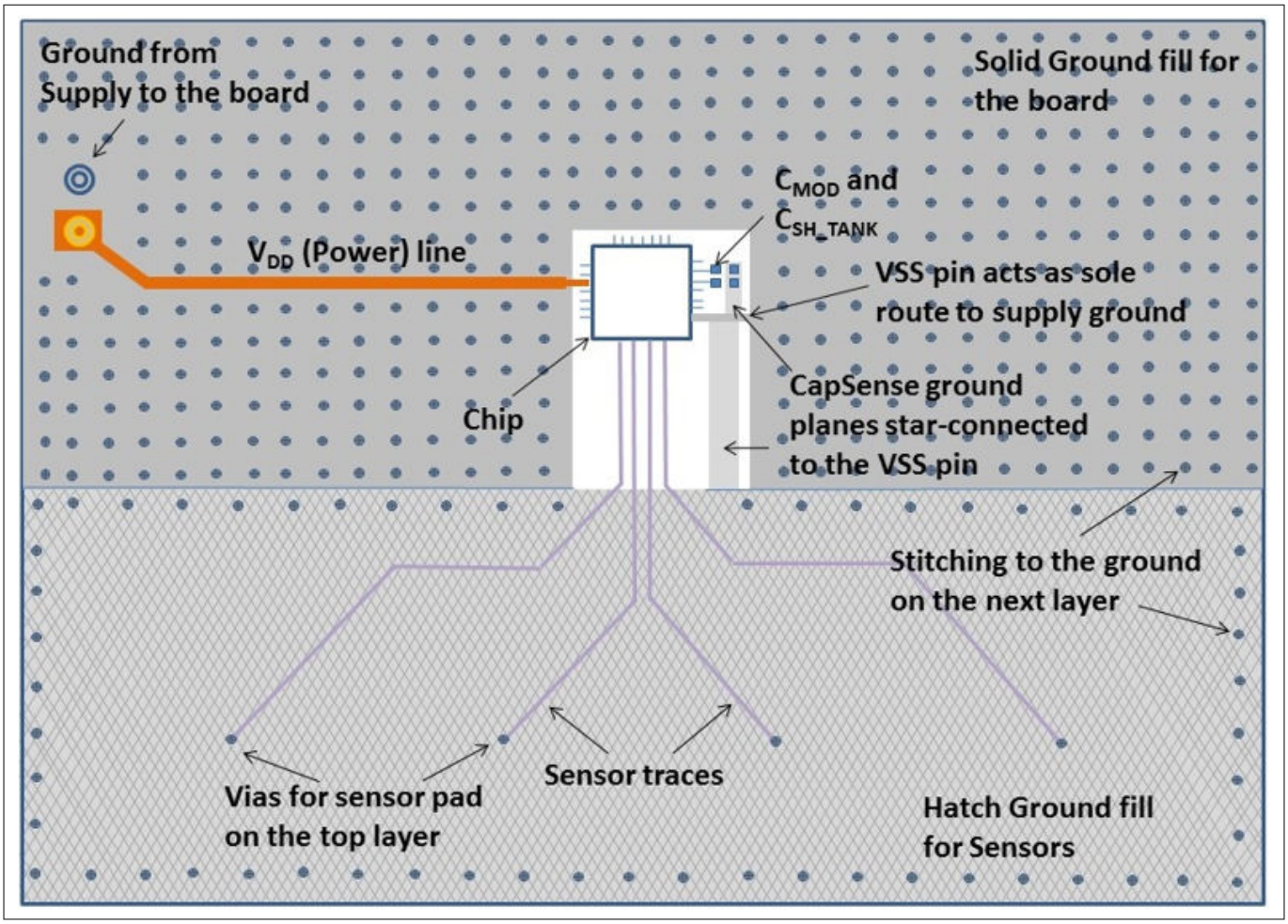


図 154　　E-pad がないチップを使用する場合の PCB 最下層のレイアウト

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を従ってください。

- 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 5 x 5 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図 155 および図 156 に示します。

7 設計上の注意事項

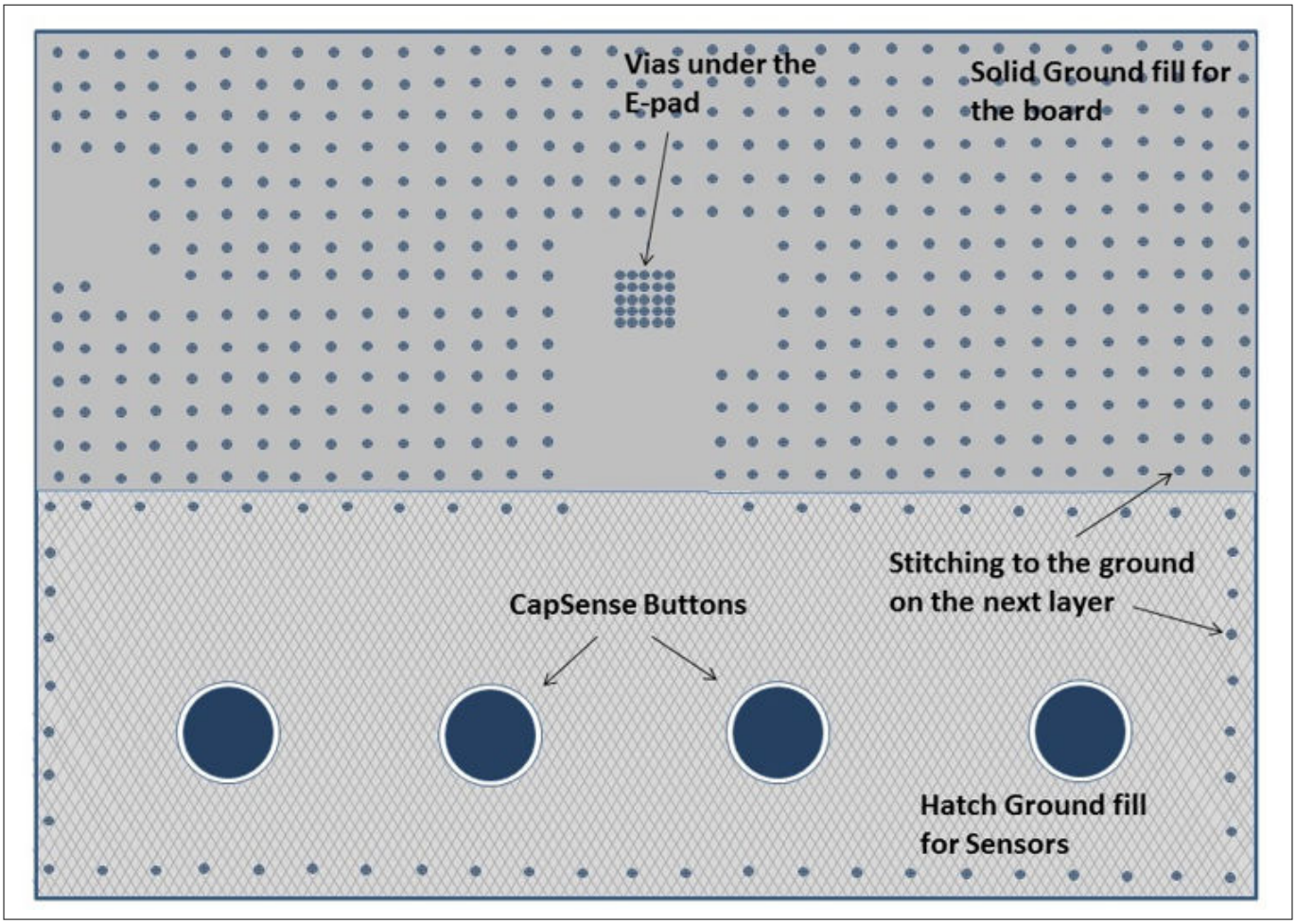


図 155　　PSOC™ 4 Bluetooth® LE を使用する場合の PCB 最上層レイアウト (E-pad あり)

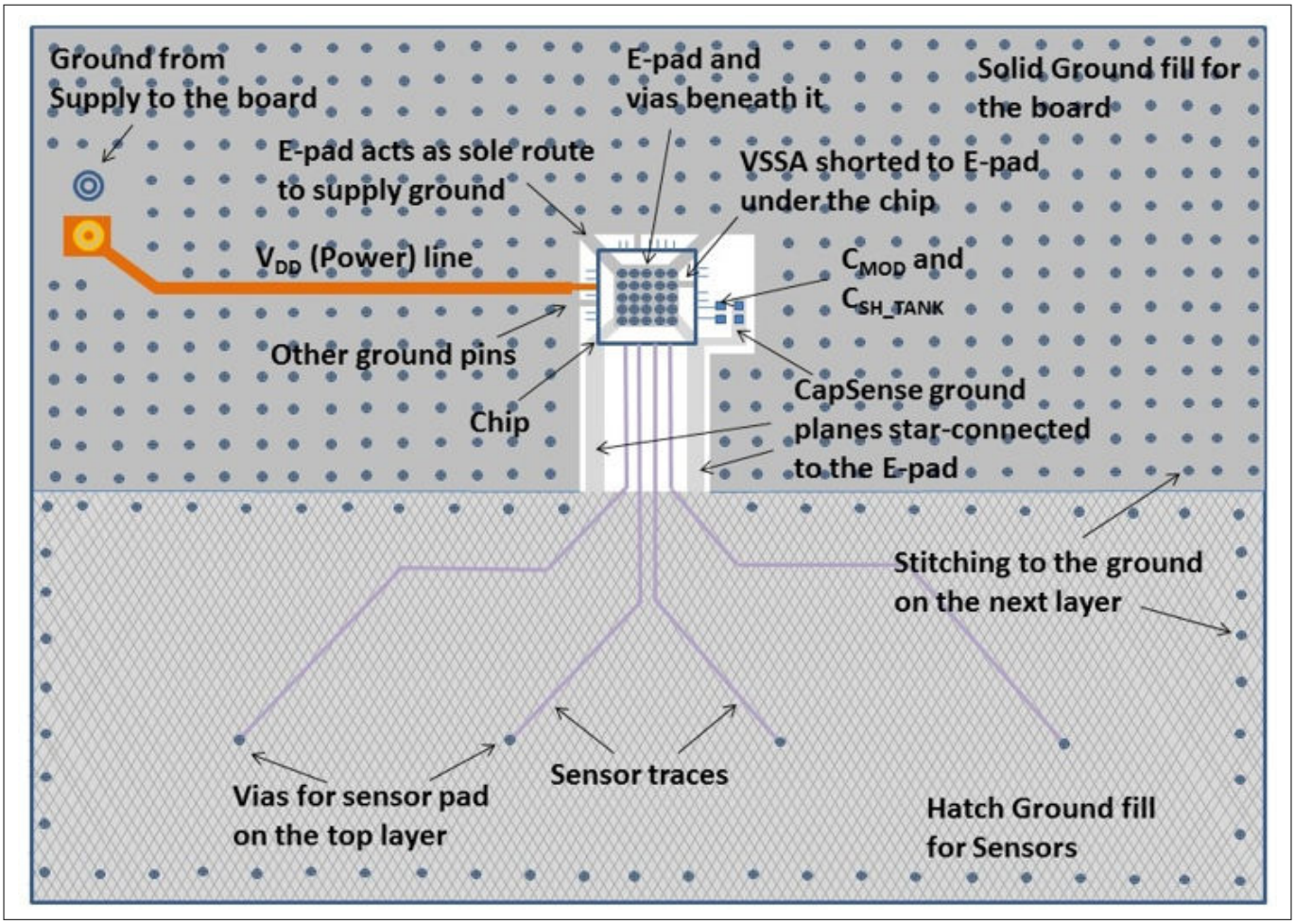


図 156　　PSOC™ 4 Bluetooth® LE を使用する場合の PCB 最下層レイアウト (E-pad あり)

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。高感度の CAPSENSE™設計で低ノイズを達成するためには、表 37 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。

表 37　　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™ 4 S シリーズ, PSOC™ 4100S Plus, PSOC™ 4100S Max
	デバイス データシートを参照してください。	VSSD	PSOC™ 4000T

(続く)

表 37 (続き) 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
VDDA[1)]	0.1µF および 1µF (バッテリで動作する場合)	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

1) PSOC™ 4 S 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薦されますが、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。

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 コンデンサの詳細についてはデバイス 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™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。

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、図 157 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の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。

7 設計上の注意事項

- 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。

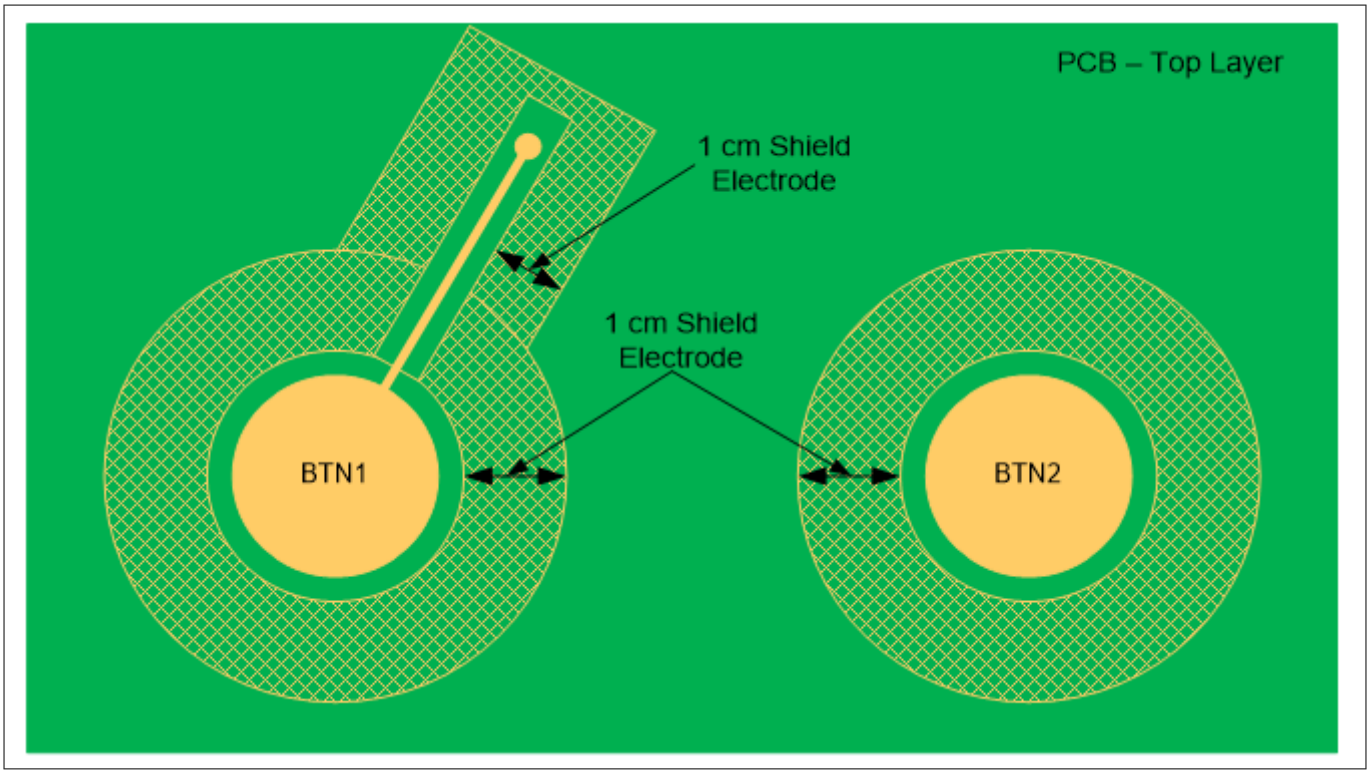


図 157　　センサー配線が最上層と最下層で配線されるときのシールド電極の配置

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。

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: V_{DD} 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 センサーは図 158 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。

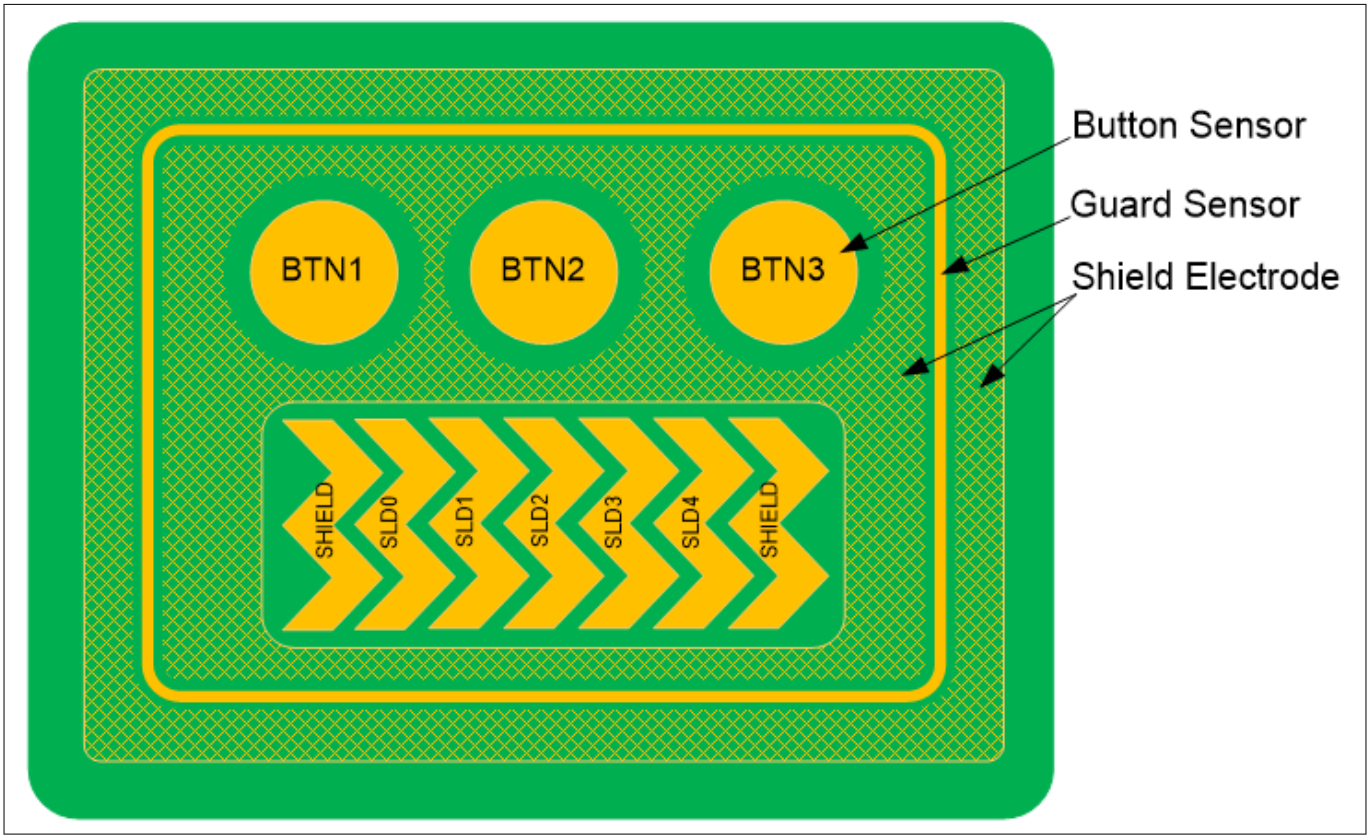


図 158 シールド電極とガード センサーを実装した PCB レイアウト

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ー 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。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。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

図 159 に示すように、複数のアプリケーションで、向上した ESD 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。グランド リングは、リダイレクトで説明されているように、ESD をリダイレクトするために使用されます。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。

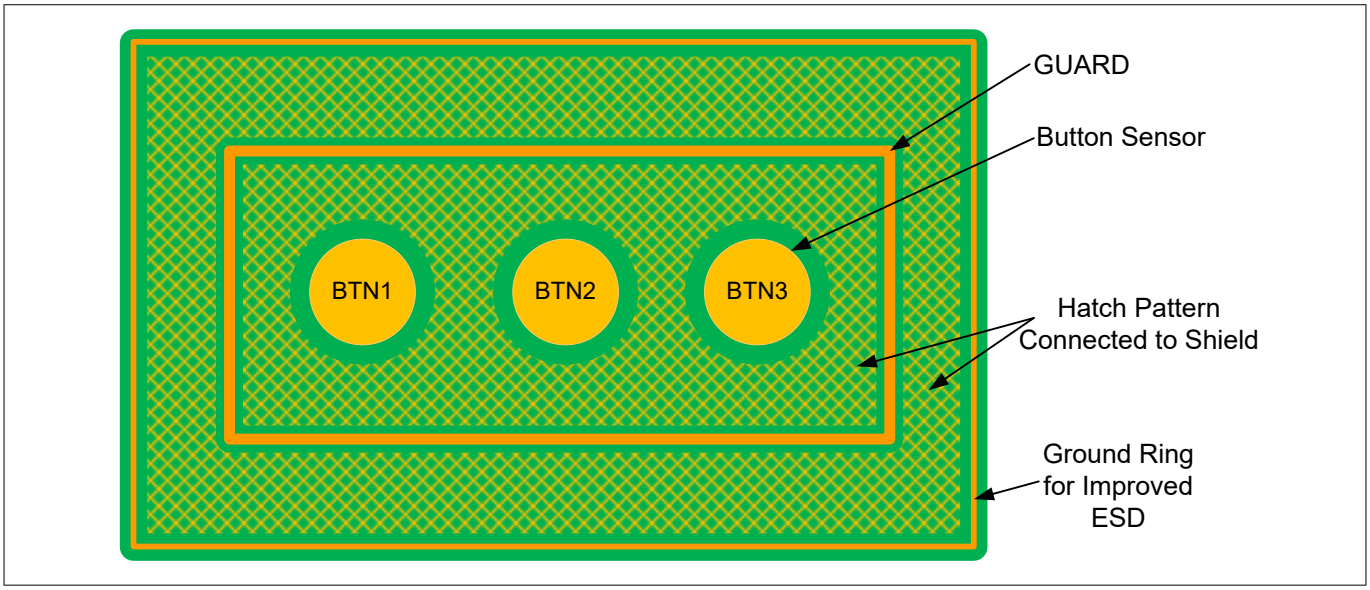


図 159 向上した ESD 性能用のグランド リング付き CAPSENSE™設計

7.4.13 回路図のルール チェックリスト

表 38 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。

表 38 回路図のルール チェックリスト

No.	区分	推奨事項/備考
1	C_{MOD}	2.2 nF。C_{MOD} は X7R または NP0 のコンデンサが使用できます。温度範囲が周囲温度よりかなり高い(つまり 50°C以上)アプリケーションでは、X7R タイプの C_{MOD} は、raw カウントの変動が非常に大きくなり、おそらく飽和します。したがって、予想される動作温度範囲に応じて C_{MOD} コンデンサのタイプを選択してください。ピンの選択は表 39 を参照してください。
2	C_{SH_TANK} [1)]	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} の使用の詳細については、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。 ピンの選択は表 39 を参照してください。
3	C_{INTA}/C_{INTB} [2)]	470pF。ピンの選択は表 39 を参照してください。

(続く)

表 38 (続き) 回路図のルール チェックリスト

No.	区分	推奨事項/備考
4	入力ラインの直列抵抗	自己容量では 560Ω、相互容量では 2kΩ。詳細については、CAPSENSE™ピンの直列抵抗を参照してください。
5	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^2C 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。詳細については、センサー ピンの選択のセクションを参照してください。
6	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	シールドラインの直列抵抗	シールドラインでは 560Ω を推奨します。

1) これらの外付けコンデンサは、第 3 世代および第 4 世代の CAPSENSE™の場合にのみ使用されます。
2) これらの外付けコンデンサは、第 3 世代および第 4 世代の CAPSENSE™の場合にのみ使用されます。

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} (capacitive sigma-delta (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。表 39 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。

注: 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、P4[2]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。

表 39 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]
PSOC™ 4000T	P4[2]	P4[3]

7 設計上の注意事項

表 40　　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}	$C_{INT\ B}$
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[1)]	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]		
PSOC™ 4000T	P4[2],	P4[3]	該当なし	該当なし

1) PSOC™ 4100S Max でサポートされるピン配列: CMOD1/CMOD2 = P4[0] /P4[1] または P4[2] /P4[3] または P7[0] /P7[1] または P5[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。

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。
  - 表 41 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, I^2C, 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。表 42 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、図 161 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。

表 41 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

表 42 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	VBACKUP	無し*	スイッチング周波数 < 8MHz
P1	VDDD	無し*	スイッチング周波数< 1MHz, SLOW スルーレート

(続く)

表 42 (続き) 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
P2, P3, P4	VDDIO2	無し*	スイッチング周波数 < 25MHz
P5, P6, P7, P8	VDDIO1	有り	推奨しません
P9, P10	VDDIOA	無し*	スイッチング周波数< 1MHz, SLOW スルーレート
P11, P12, P13	VDDIO0	無し*	スイッチング周波数 < 80MHz
P14	VDDUSB	無し*	該当なし

注: *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。図 160 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。*

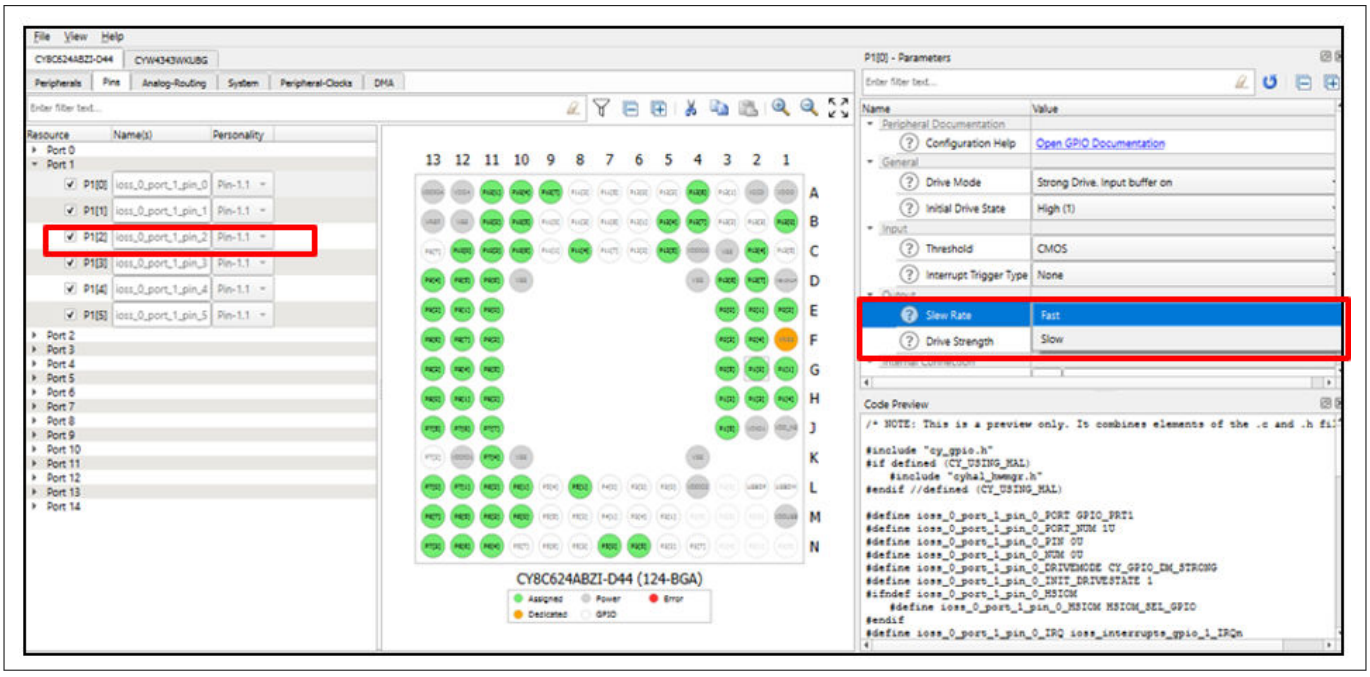


図 160 GPIO のスルーレートの選択

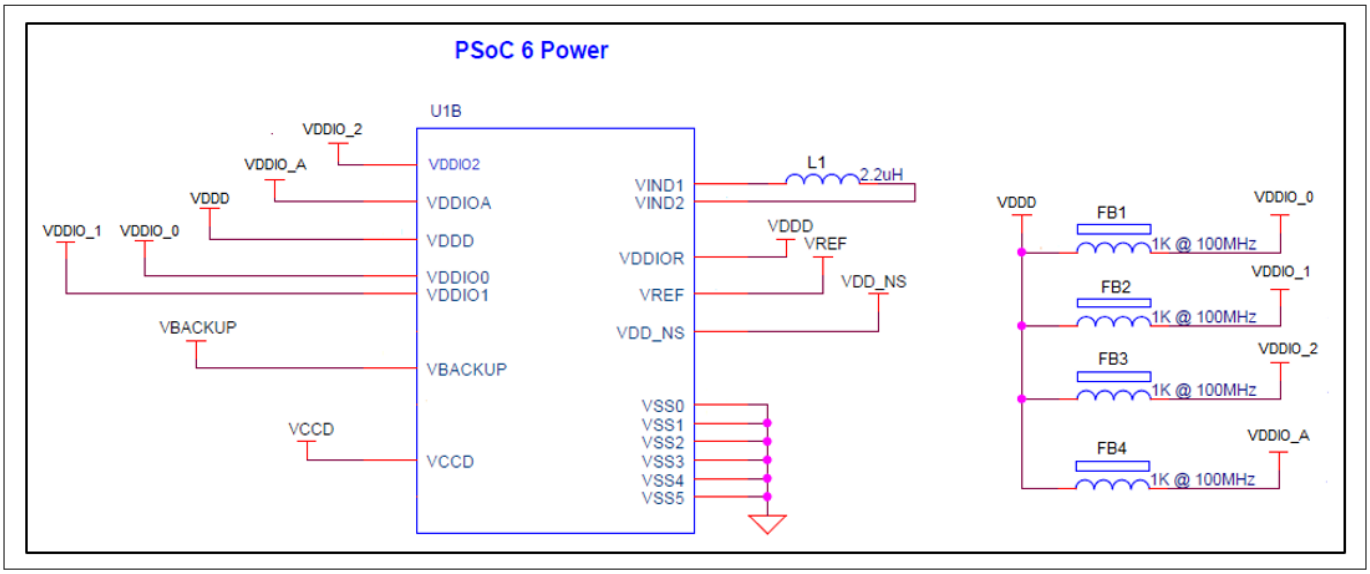


図 161　　外部で分離された供給ドメイン

7.4.14　レイアウト ルール チェックリスト

表 43 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。

表 43　　レイアウト ルール チェックリスト

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	該当なし	円形または角が丸い長方形です。
		サイズ	5 mm	15 mm	10 mm
		グランド ハッチまでの間隔	0.5 mm	2 mm	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	1.5 mm	8 mm	8 mm
		セグメント間の間隔	0.5 mm	2 mm	0.5 mm
		セグメントの高さ	7 mm	15 mm	12 mm
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	5 mm	
		スライダーの厚さ	該当なし	5 mm	
		タッチパッドの厚さ	該当なし	0.5 mm	
4	センサー配線	幅	該当なし	7 mil	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。

(続く)

表 43 (続き) レイアウト ルール チェックリスト

No.	区分		最小値	最大値	推奨事項/備考
		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	0.25 mm	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。
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	10 mil
6	グランド	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図 162 を参照してください。
8	シールド電極	広さ	該当なし	1 cm	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	グランド リング (ESD 保護)	形状	該当なし	該当なし	角が丸いボード周辺部。ソリッドトレースでもハッチフィルでも可。
		厚さ	該当なし	該当なし	液滴によってセンサーの誤作動を引き起こす可能性があるため、最上層では推奨されません。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) のリングを使用してください。

(続く)

表 43 (続き) レイアウト ルール チェックリスト

No.	区分		最小値	最大値	推奨事項/備考
11	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} の配置例は図 162 を参照してください。
12	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} の配置例は図 162 を参照してください。
13	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} の配置例は図 162 を参照してください。
14	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} の配置例は図 162 を参照してください。

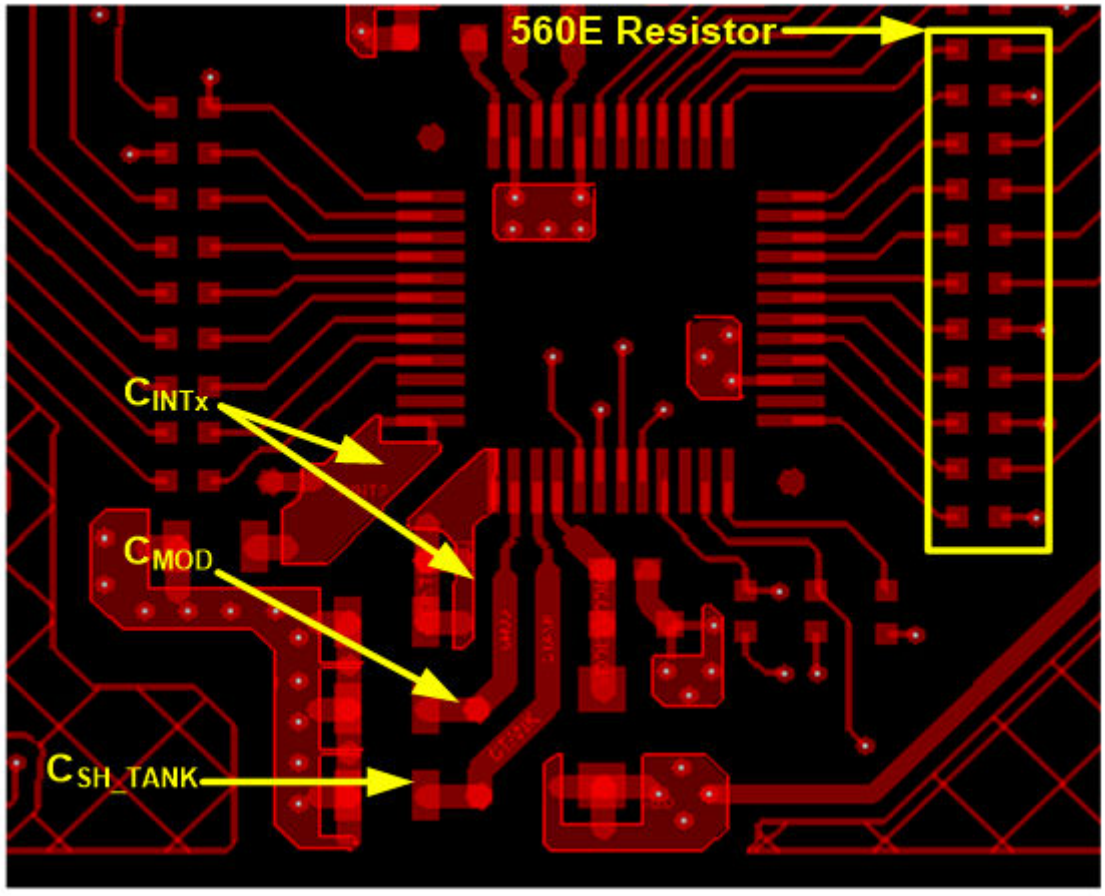


図 162 PSOC™ 4200M デバイスにおける入力ライン上の C_{MOD}, C_{INTx}, C_{SH_TANK}, および直列抵抗の配置例

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。図 163 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、バッテリ電源アプリケーションでは発生しません。

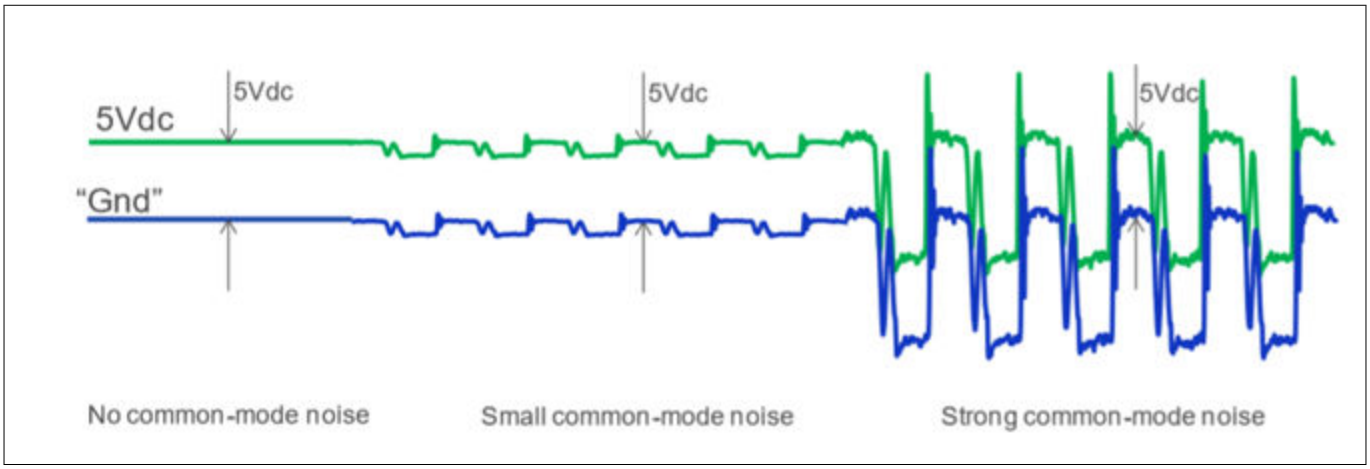


図 163 電源のコモンモードノイズ

***注**: 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。

1. 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。

   図 164 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。

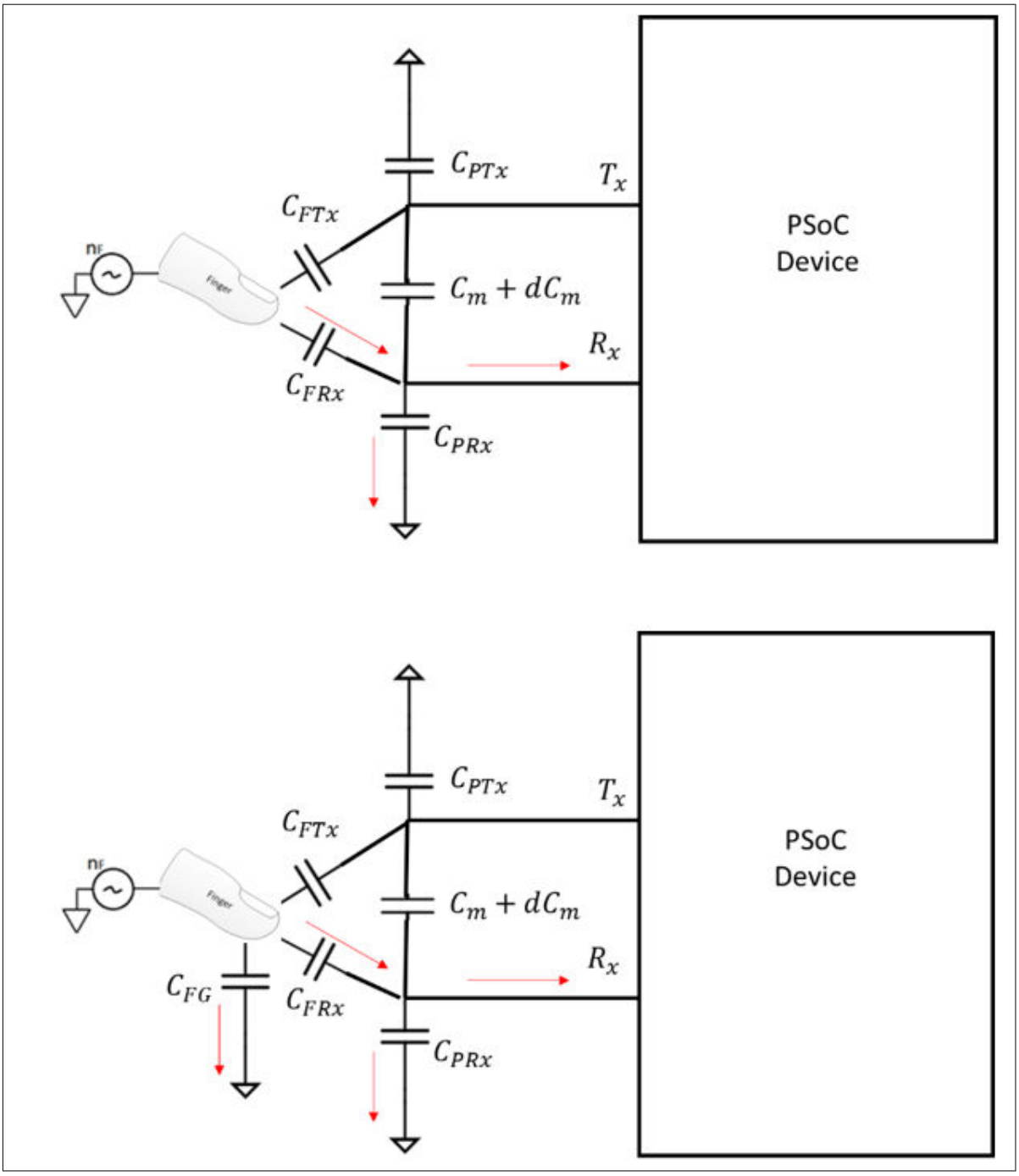


図 164　　Finger injected ノイズに対する接地の影響

2. 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。
3. 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。
4. 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。

1. 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。
2. 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。第 5 世代および第 5 世代 低消費電力 CAPSENSE™デバイスは、以前の世代の CAPSENSE™デバイスと比較して、電源リップルに対してより堅牢です。電源リップルがデータシートに記載されている範囲内であれば、これらのデバイスを使用した設計に LDO は不要である可能性があります。しかし、システム全体の性能要件が満たされているかどうかは検証し、確認することを推奨します。
3. 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。
4. 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。
5. 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。表 44 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)。

表 44 ESD 保護のためのオーバーレイの厚さ

素材	ブレークダウン電圧 (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
フォーマイカ	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

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。図 165 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 12 kV ESD に耐えられます。

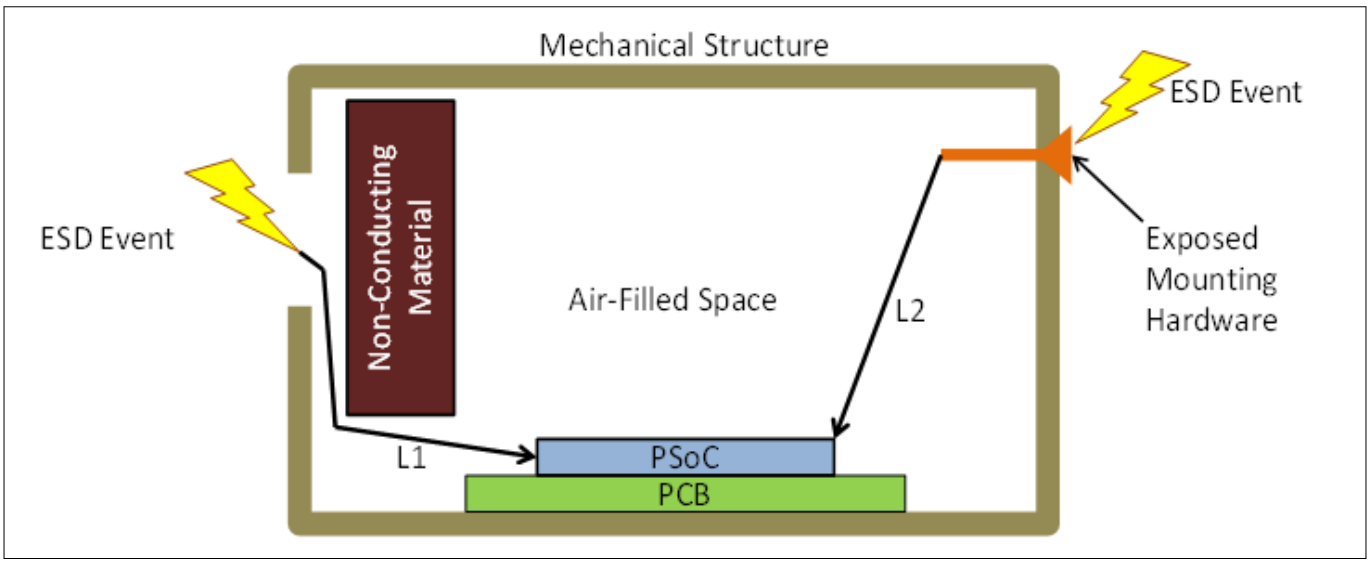


図 165 ESD パス

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表 44 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、図 166 で示すように、回路基板の周辺にグランド リングを配置します。グランド リングはシャーシ グラ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。

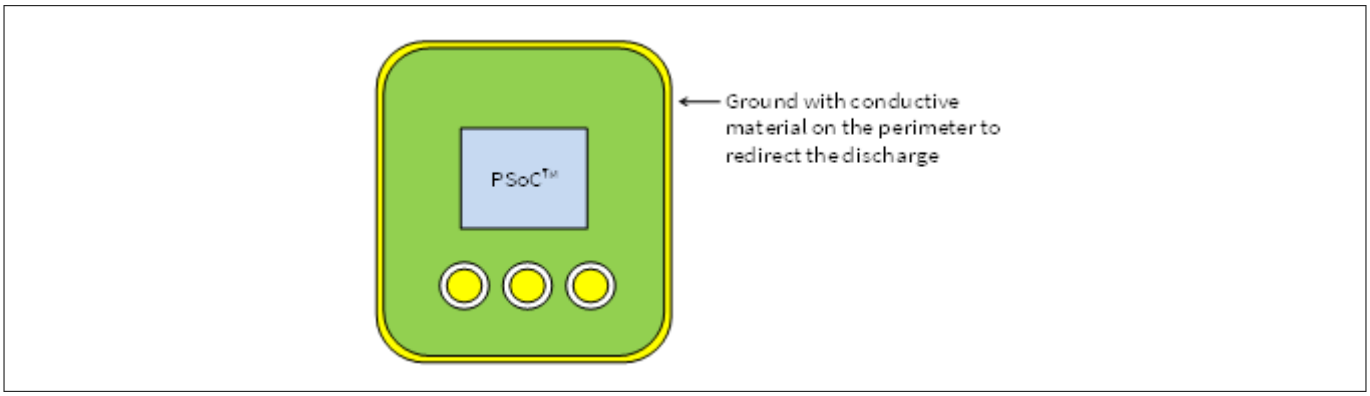


図 166 グランド リング

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 保護デバイスを表 45 に示します。

表 45 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 PRoC 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

グランド面

一般的には、PCB の適切なグランド面は 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。図 167 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。

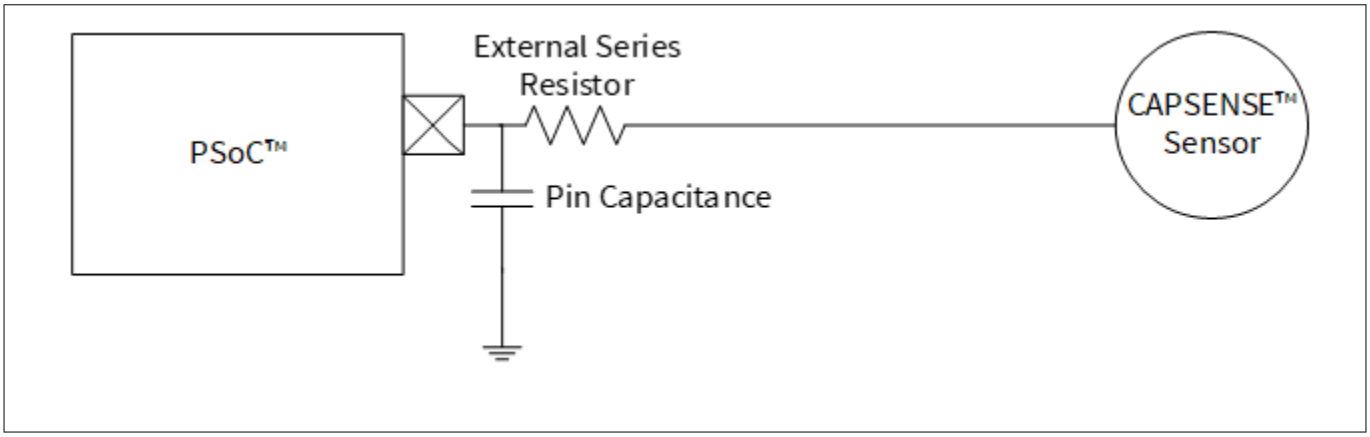


図 167　　RC フィルター

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

I^2C 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。I^2C 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、VVIL レベルが 0.3 V_{DD} の I^2C 仕様範囲内に維持されるため、I^2C の動作には影響を与えません。

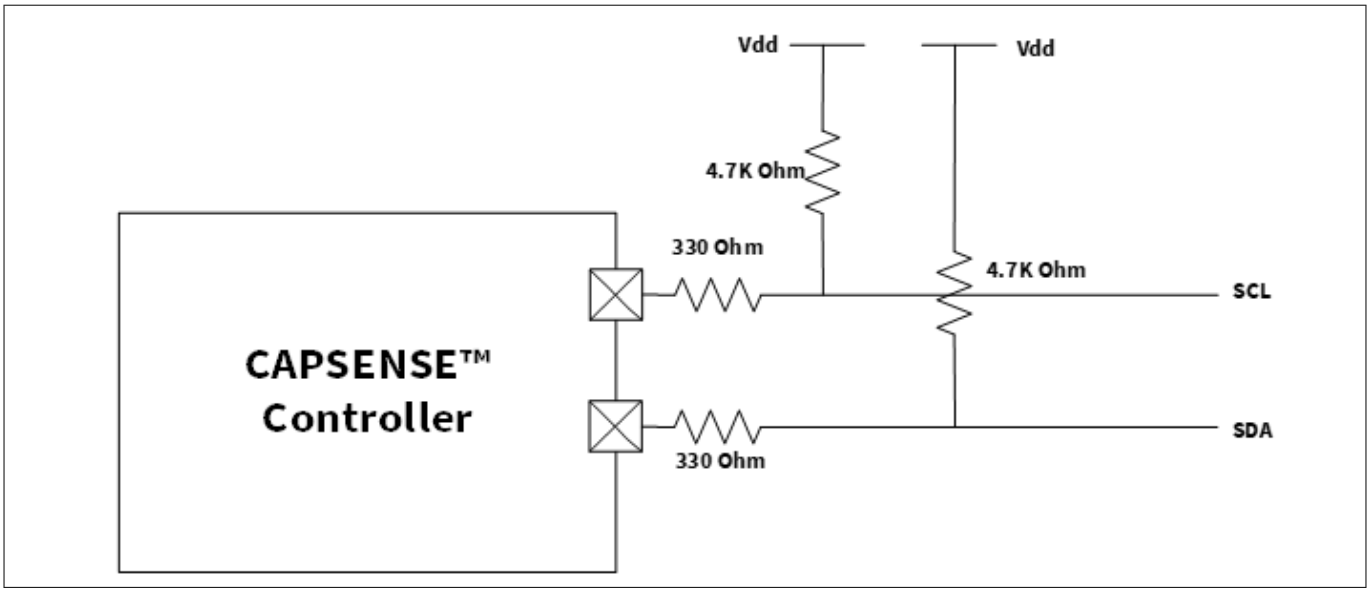


図 168　　通信ライン上の直列抵抗

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って流れ、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。

図 169 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、図 170 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。

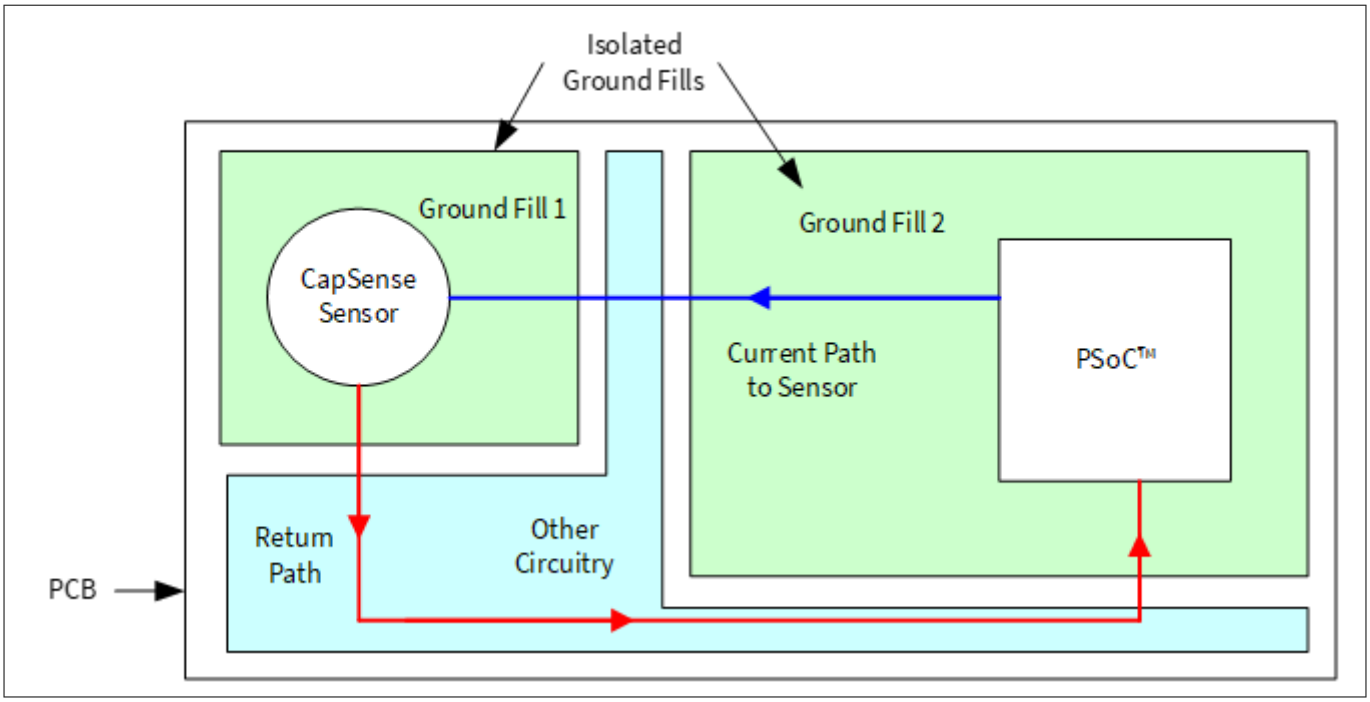


図 169　　不適切な電流ループ レイアウト

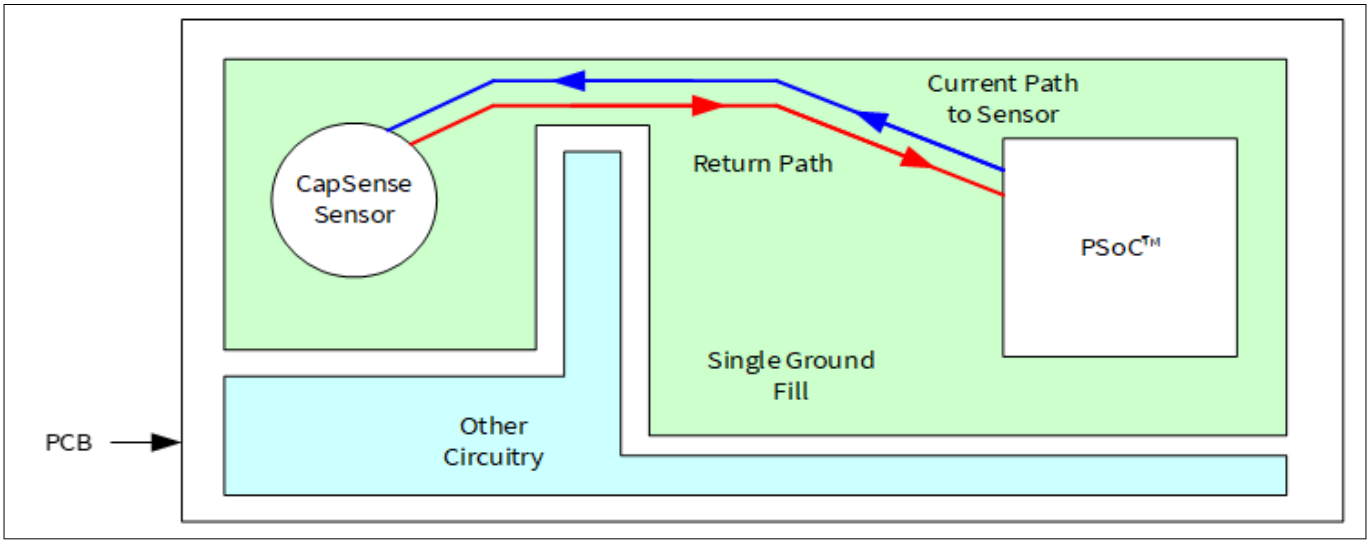


図 170　　適切な電流ループ レイアウト

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。図 171 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。

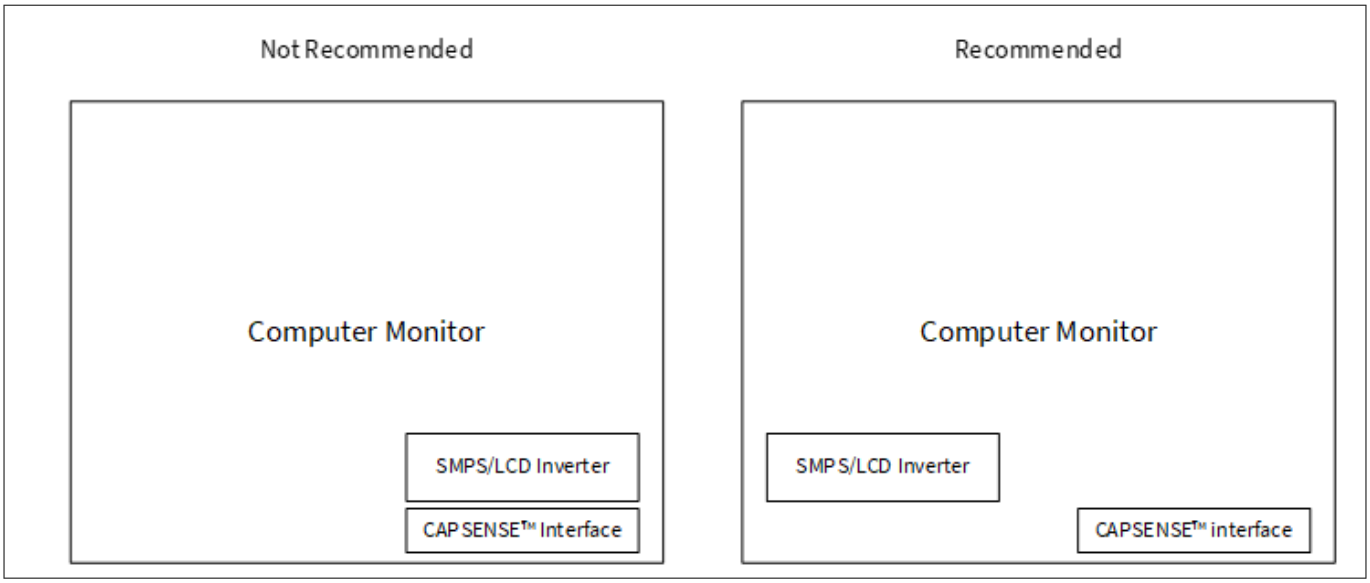


図 171　　ノイズ源からの分離

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。PSOC™は低い電圧で動作できるため、放射を低減できます。図 172 および図 173 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。

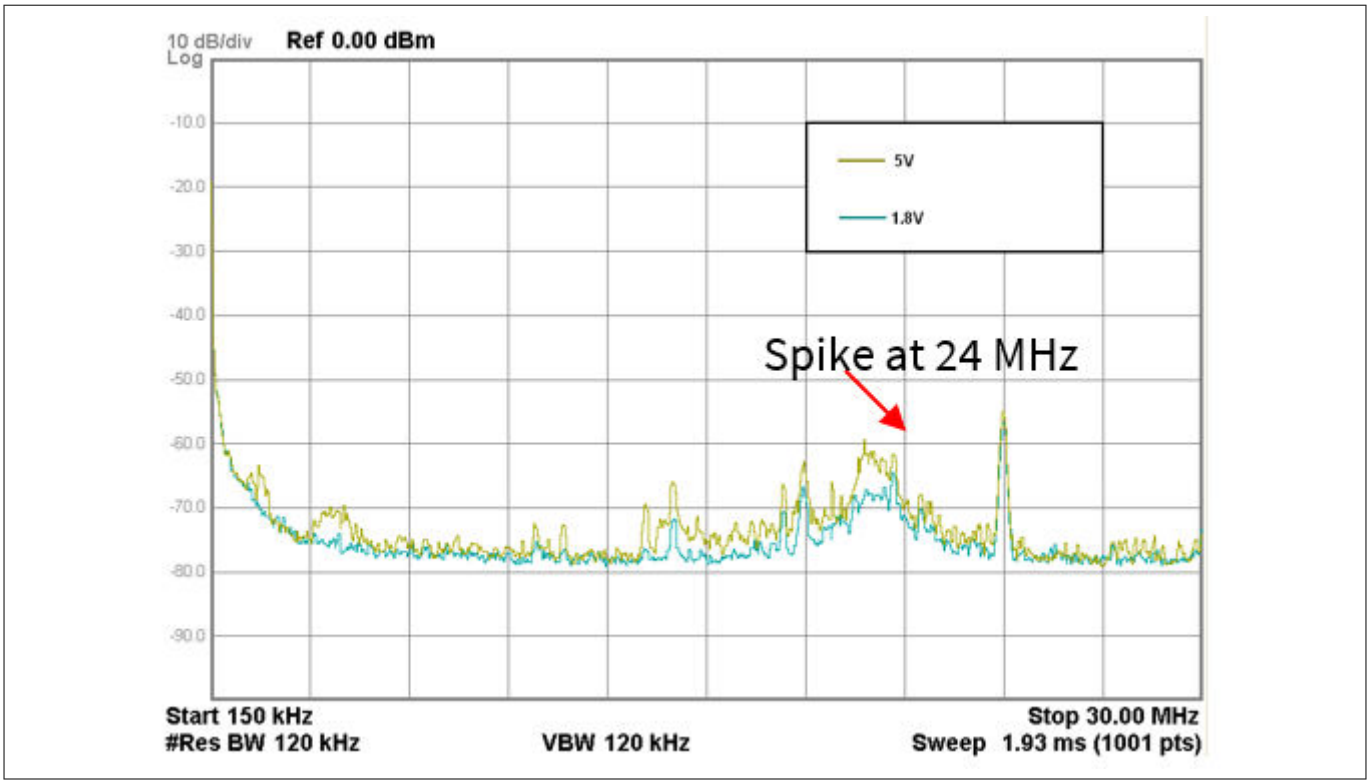


図 172　　V_{DD} の放射性妨害波への影響 (150kHz～30MHz)

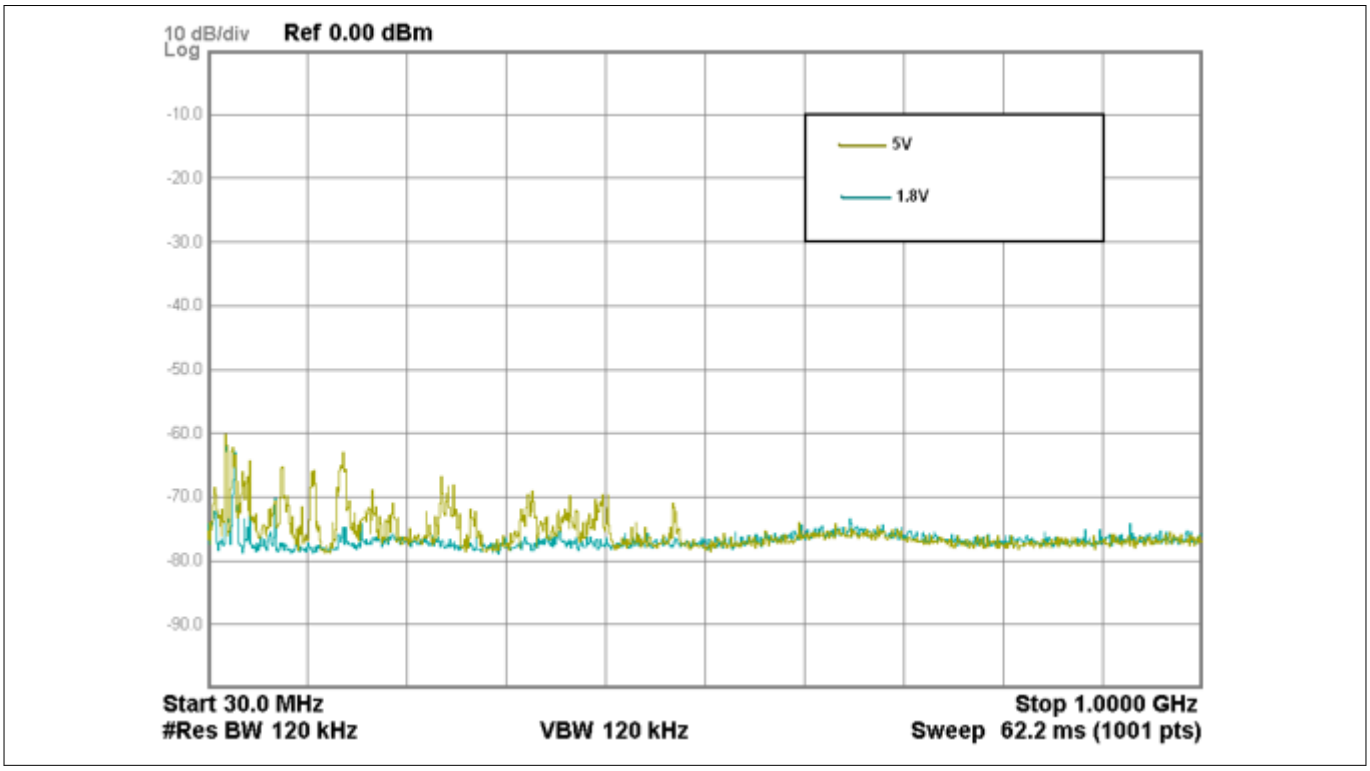


図 173　　V_{DD} の放射性妨害波への影響 (30MHz～1GHz)

***注**:　　周波数軸は対数スケールです。*

7 設計上の注意事項

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。図 174 および図 175 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。

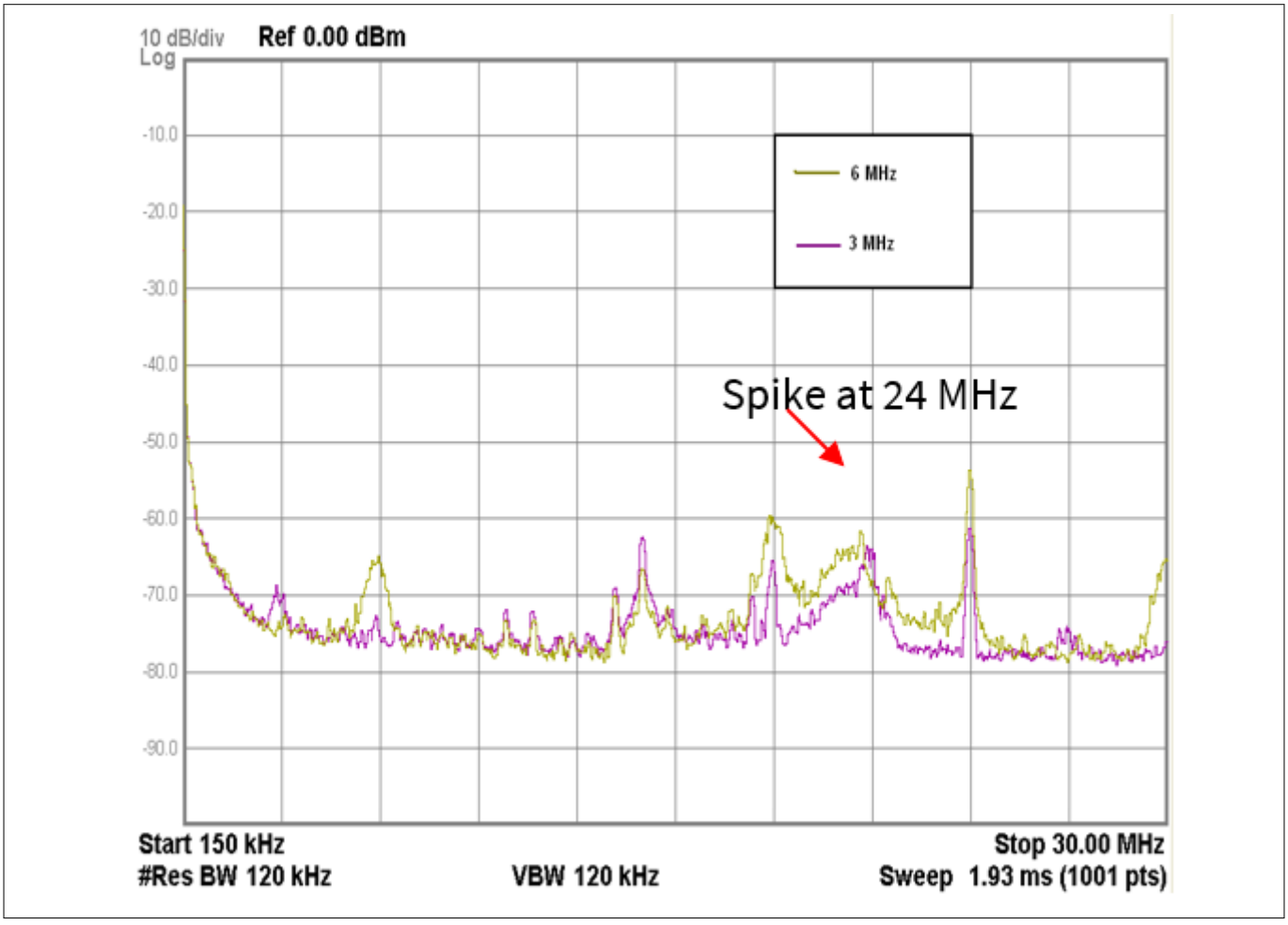


図 174 センサースイッチング周波数の放射性妨害波への影響 (150kHz～30MHz)

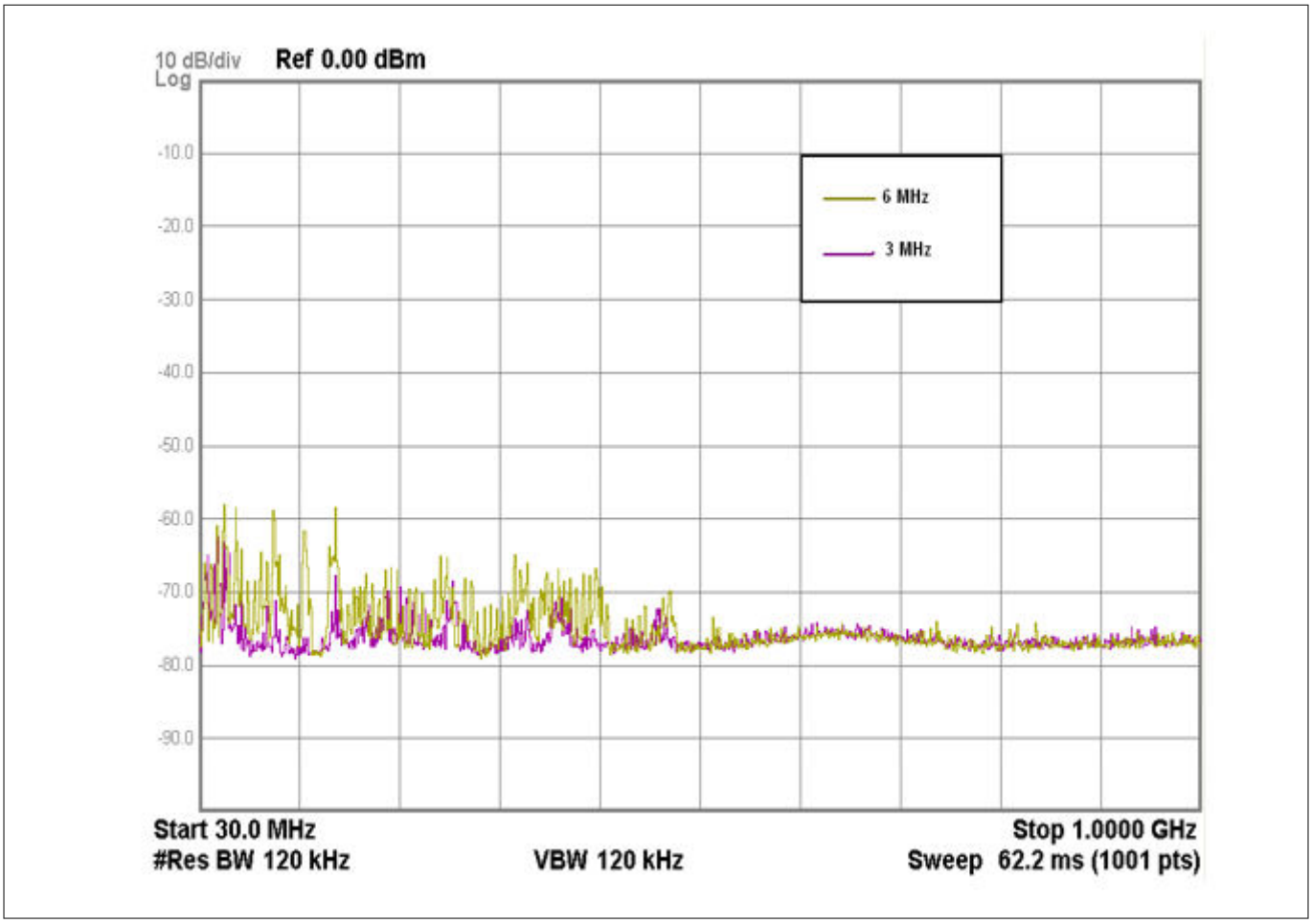


図 175　　センサースイッチング周波数の放射性妨害波への影響 (30MHz～1GHz)

注:　　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に依存します。PSOC™ 4 S シリーズ, PSOC™ 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については、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。

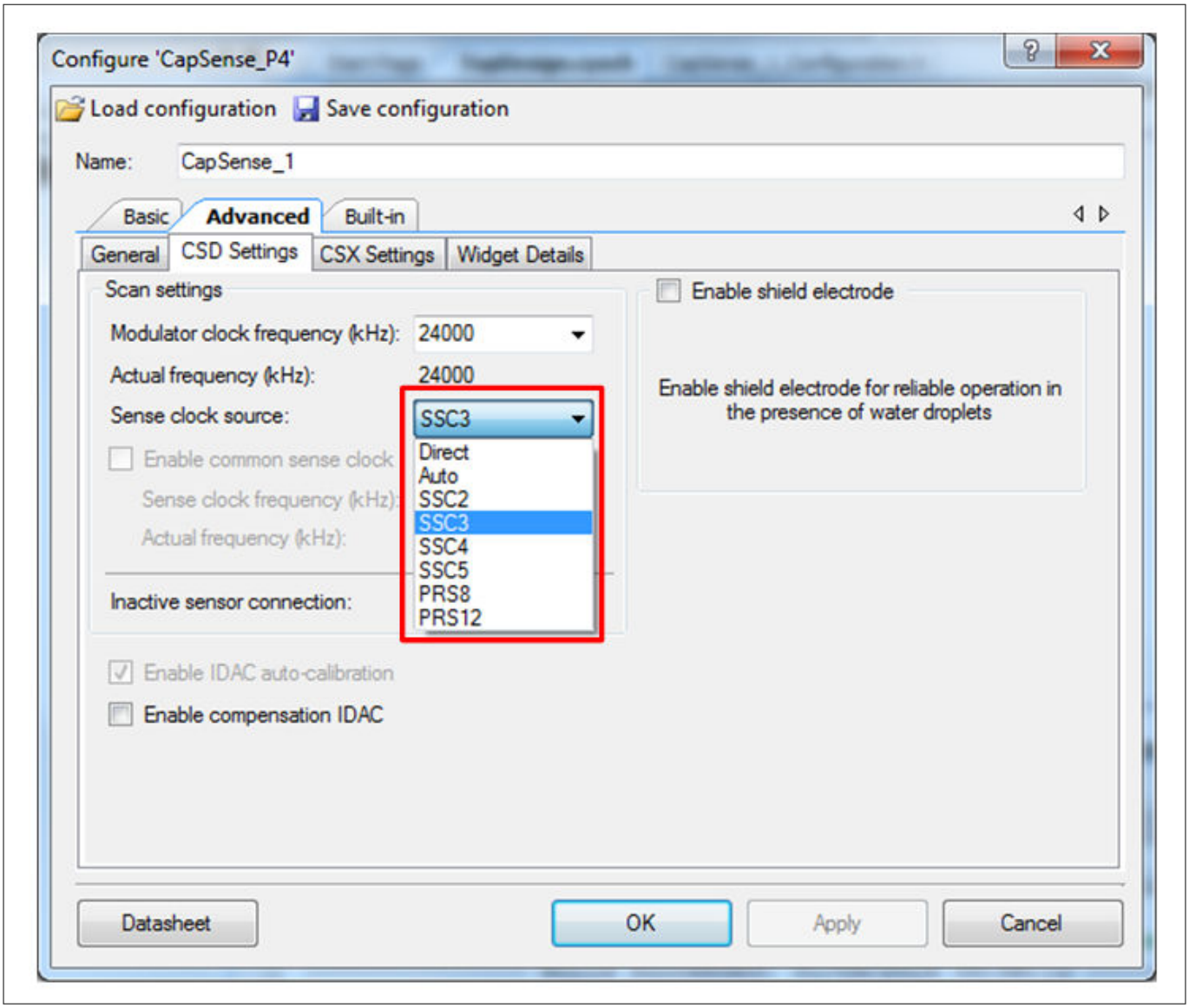


図 176　　PSOC™ 4 S シリーズ, PSOC™ 4100S Plus, および PSOC™ 4100PS ファミリのセンス クロック ソース

シールド信号

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 (図 157)。

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図 177 および図 178 に示されます。

***注**:　　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。*

7 設計上の注意事項

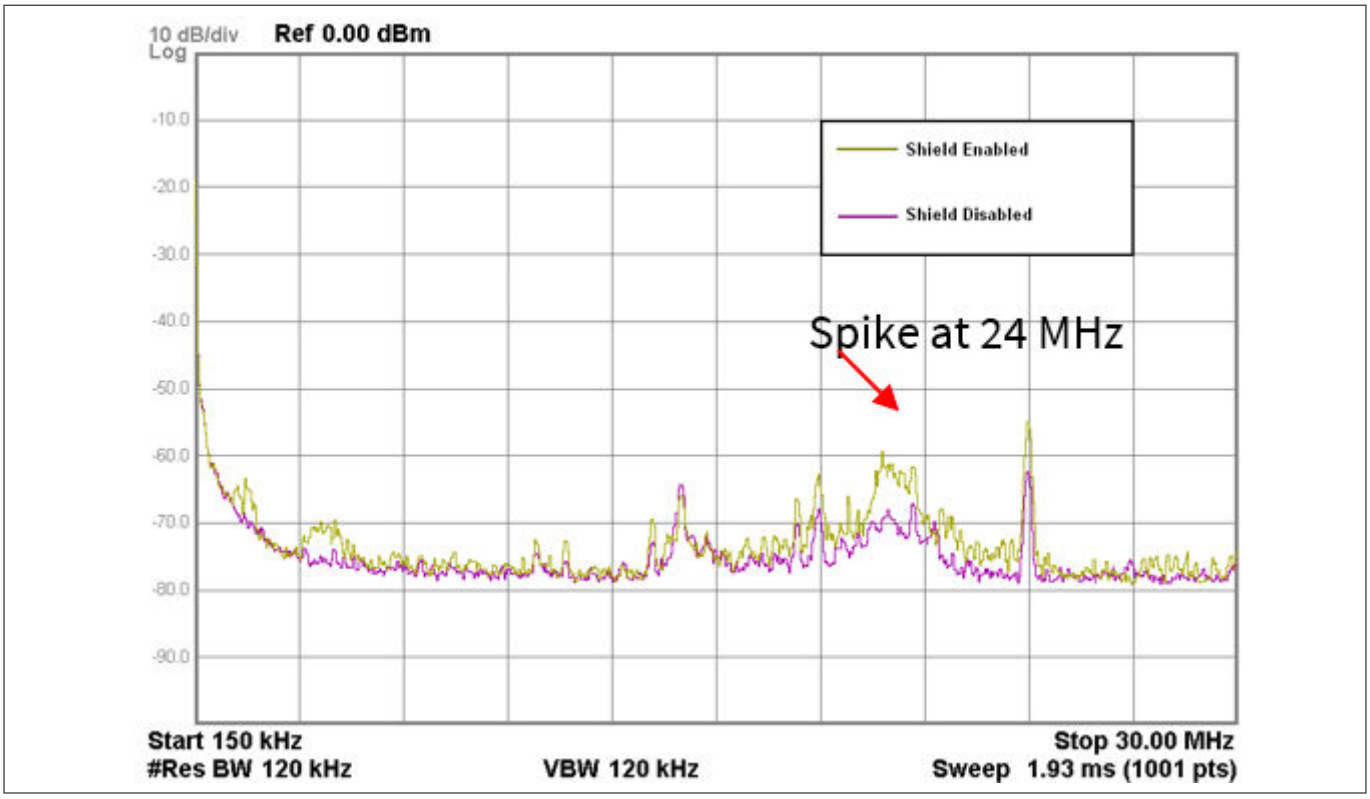


図 177　　シールド電極の放射性妨害波への影響 (150kHz～30MHz)

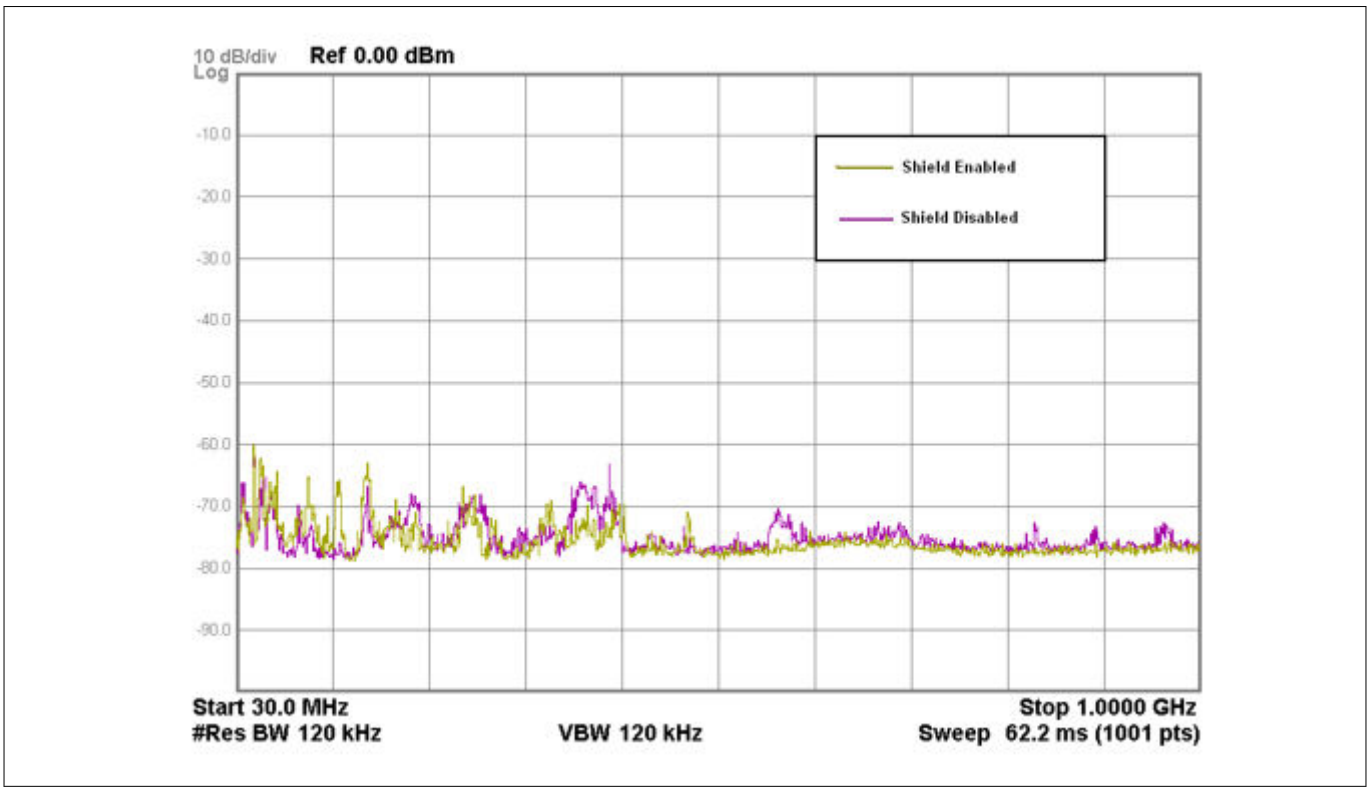


図 178　　シールド電極の放射性妨害波への影響 (30MHz～1GHz)

***注**:　　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器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器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。

図 179 および図 180 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。

表 46　　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	
	0.426 ms	**0.106 ms**
変調器クロック分周器	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	0.085 ms	0.021 ms

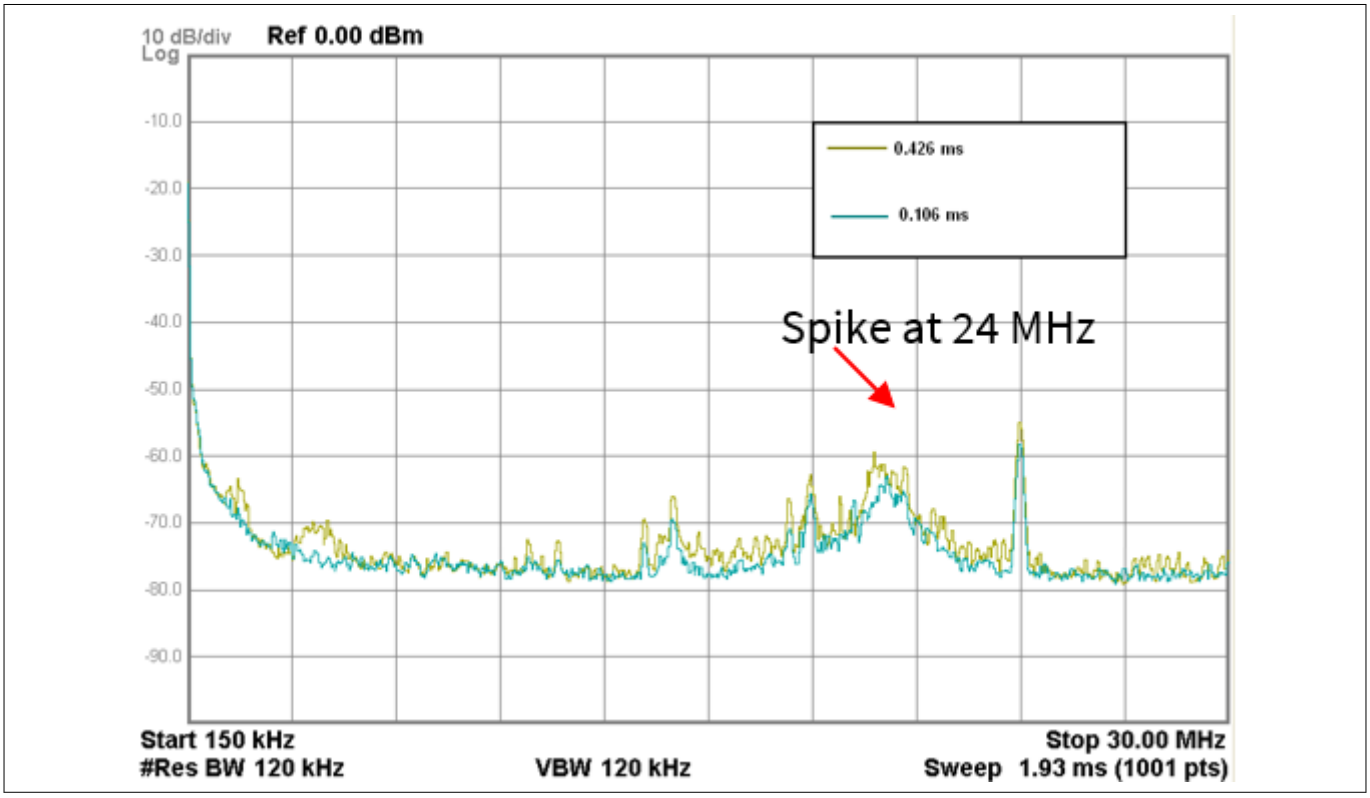


図 179　　スキャン時間の放射性妨害波への影響 (150kHz～30MHz)

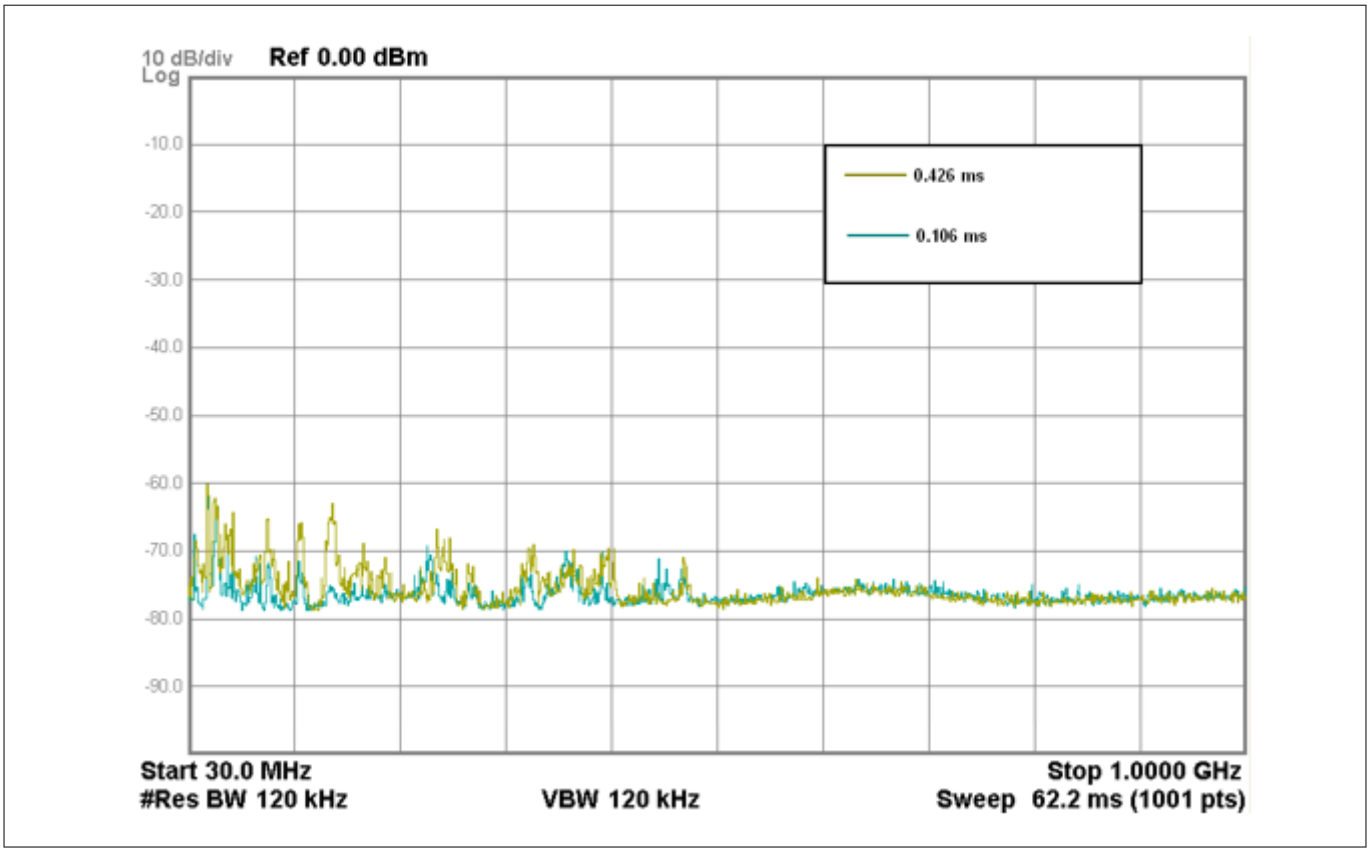


図 180　　スキャン時間の放射性妨害波への影響 (30MHz～1GHz)

注:　　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。図 181 および図 182 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。

7 設計上の注意事項

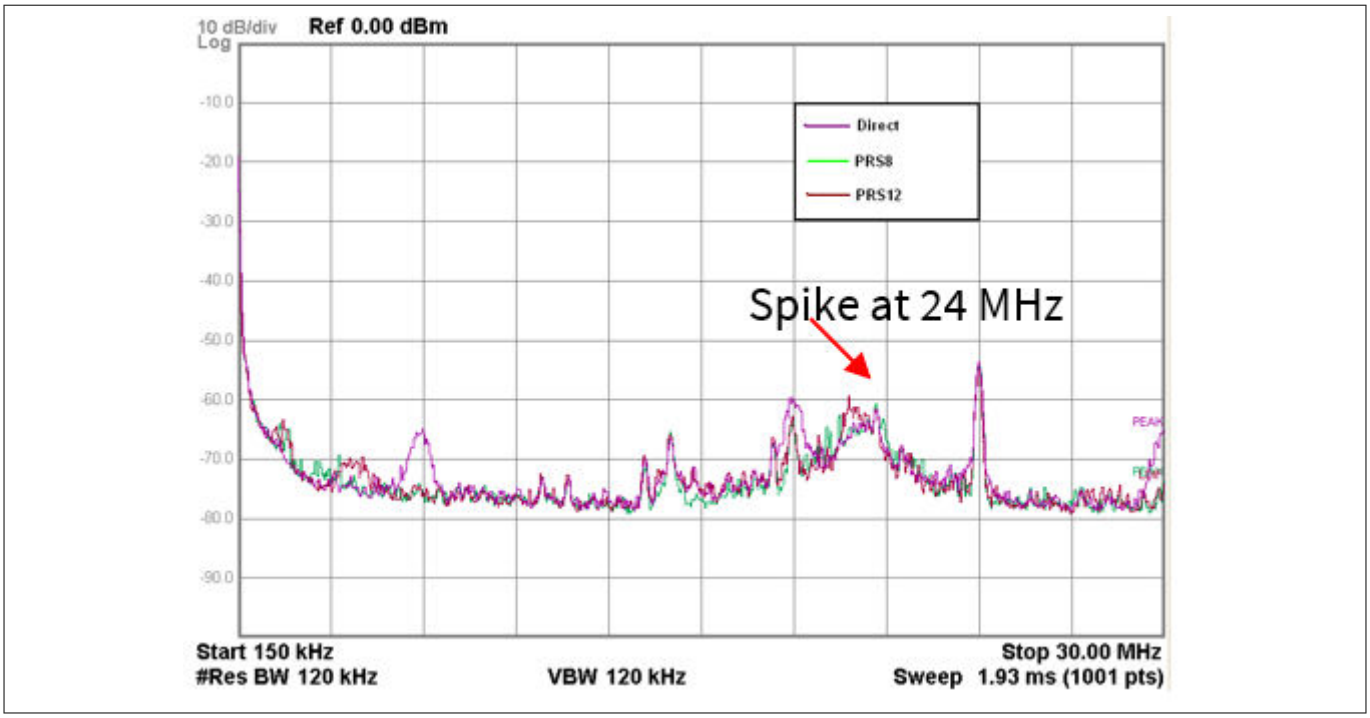


図 181　　アナログ スイッチ駆動ソースの放射性妨害波への影響 (150kHz～30MHz)

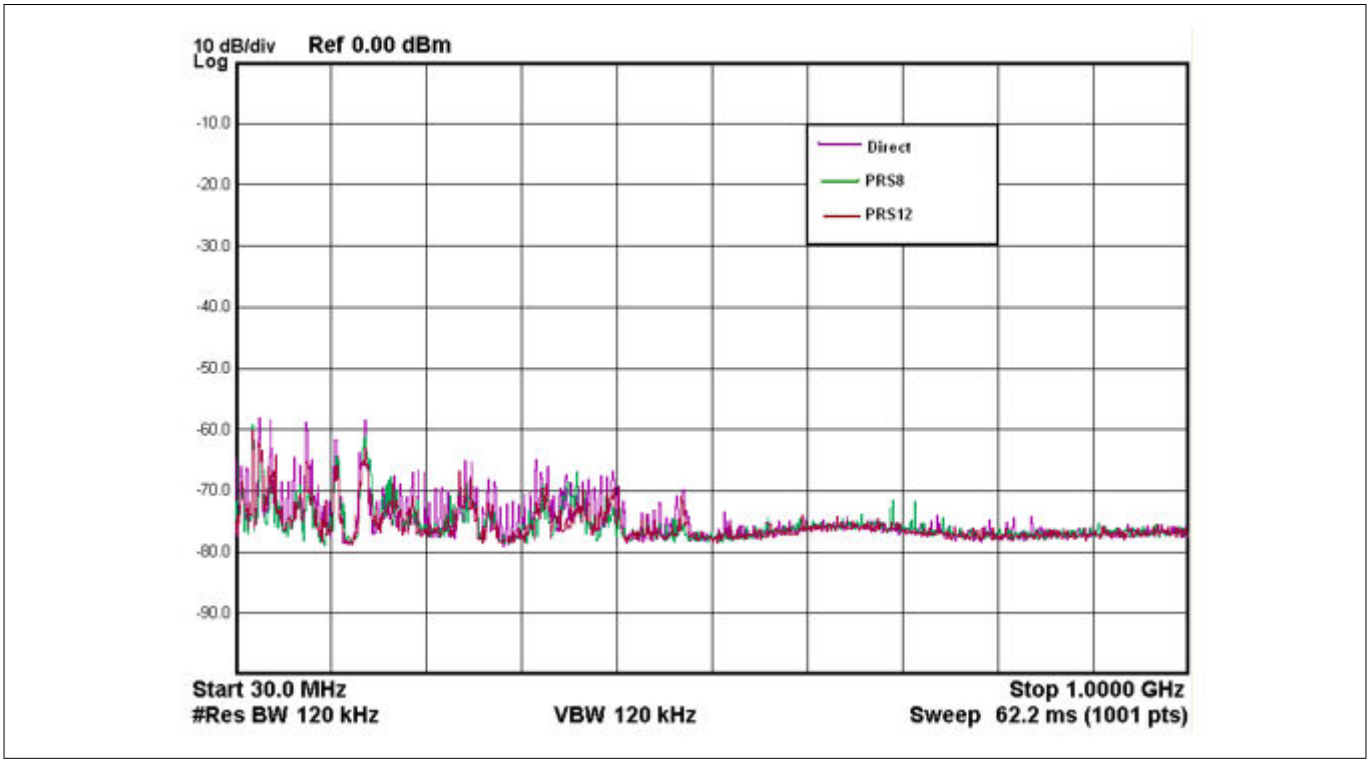


図 182　　アナログ スイッチ駆動ソースの放射性妨害波への影響 (30MHz～1GHz)

***注**:　　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。図 183 および図 184 に、異なる非アクティブなセンサー終端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。

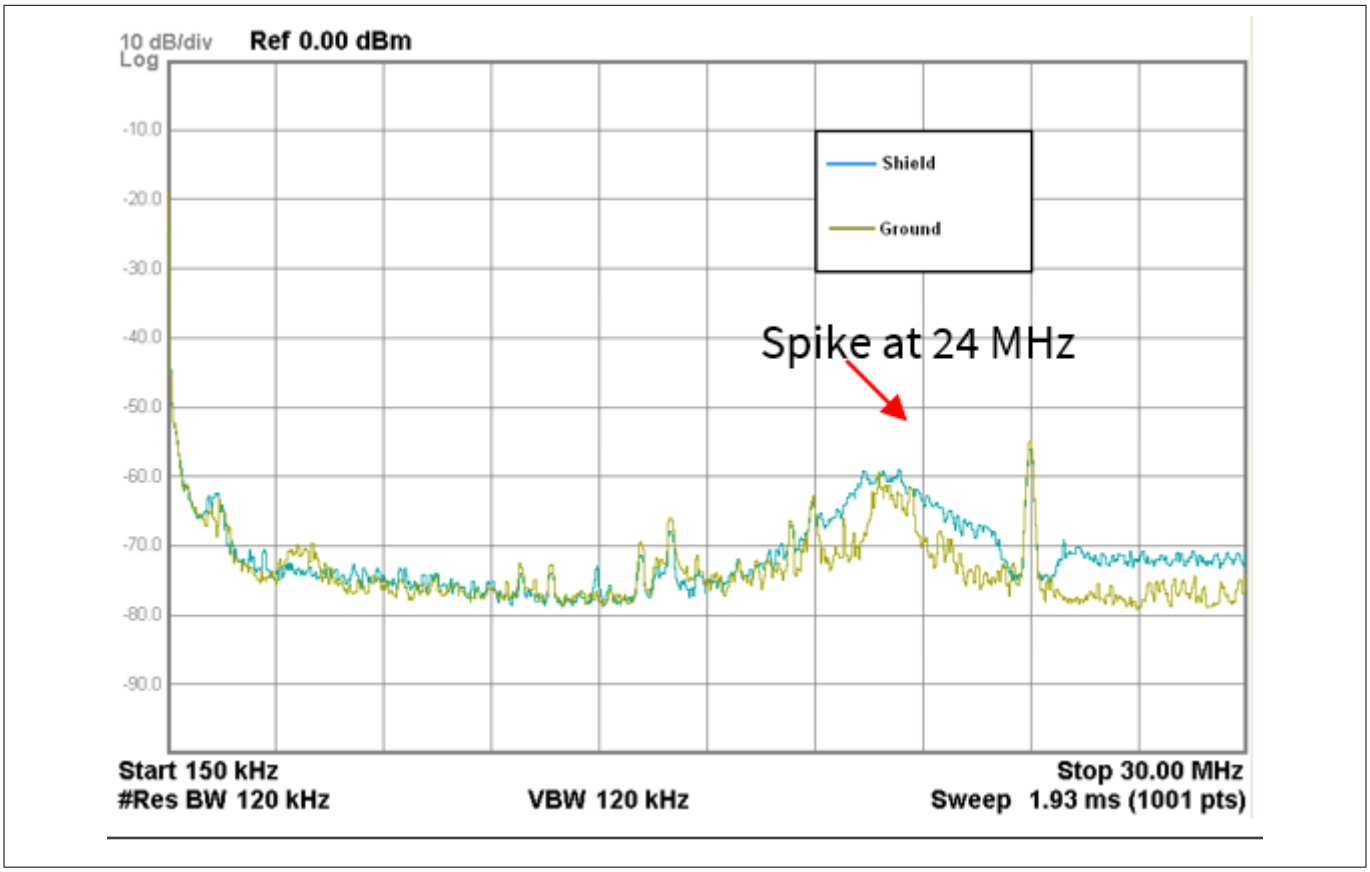


図 183　　非アクティブなセンサーの放射性妨害波への影響 (150kHz～30MHz)

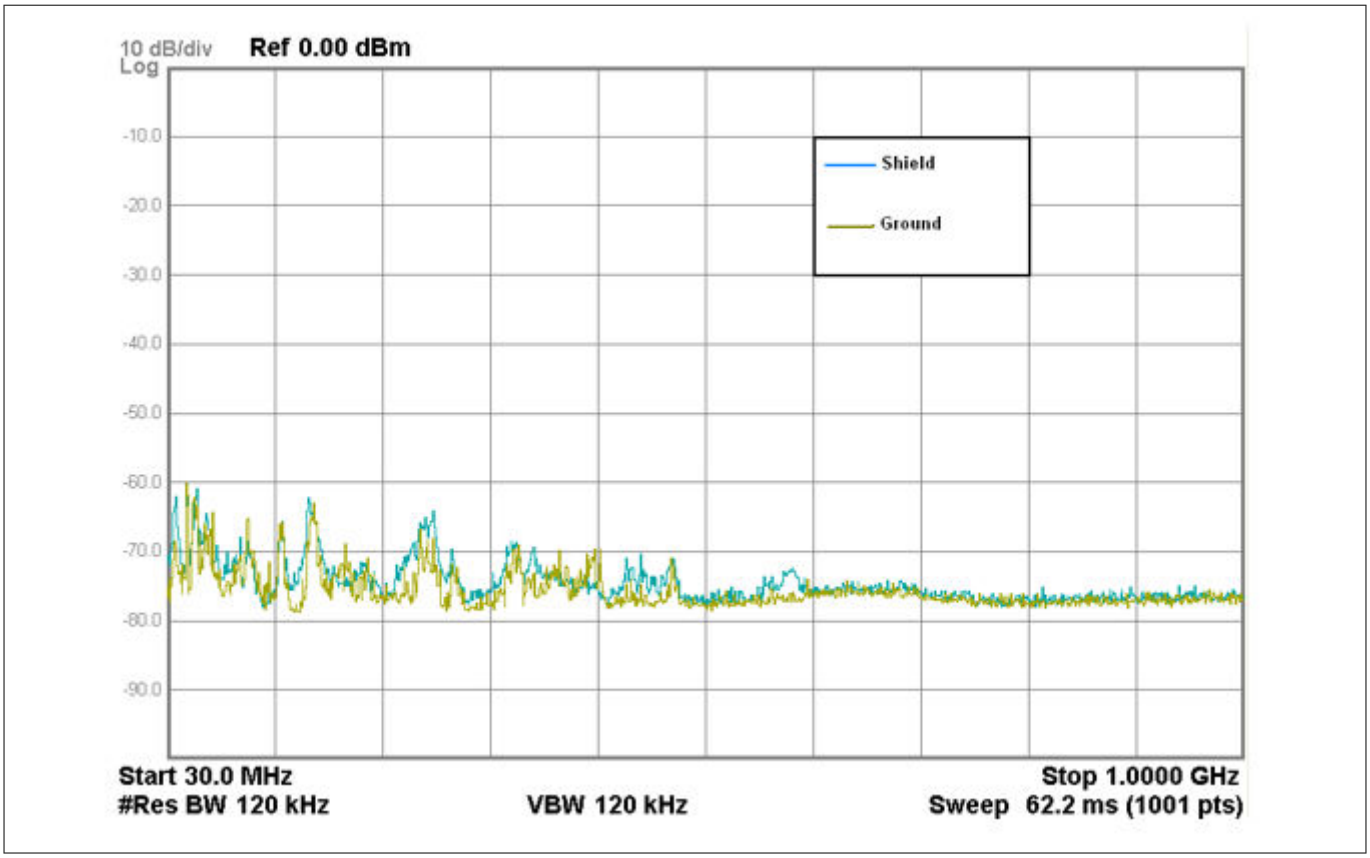


図 184　　非アクティブなセンサーの放射性妨害波への影響 (30MHz～1GHz)

注:　　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。詳細については電源レイアウトの推奨事項およびデバイス 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。

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図 185 に示します。図 185 から、IDAC (IRX) 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、$C_{bodyDG}>>C_{fs}$ と $C_{bodyDG}<<C_{fs}$ を引き起こす 2 つの極端な条件を考えてみましょう。

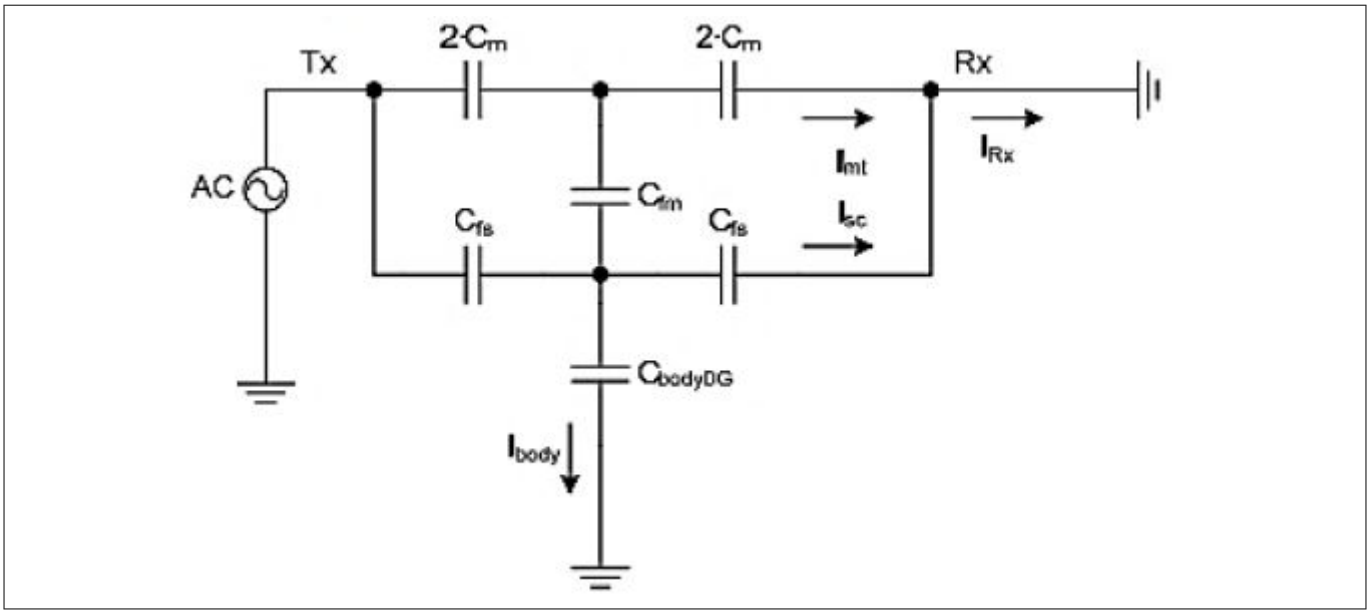


図 185　　ボタンに指を置いたときの CSX センサーの等価回路

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} = デバイスのグランドに対するボディ容量

$$I_{RX} = I_{mt} + I_{sc}$$

式 81　　CSX 方式で IDAC から引き出される電流

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

$C_{bodyDG} >> C_{fs}$ 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、図 186 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。

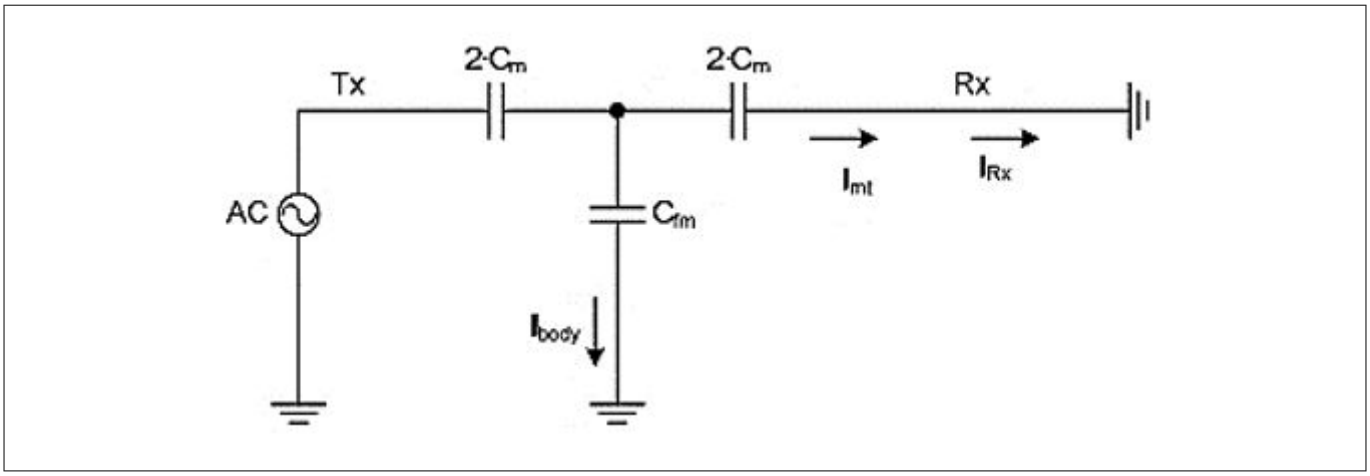


図 186　　C_{bodyDG} >> C_{fs} の場合の CSX センサーの等価回路

7.6.1.2　CbodyDG<<Cfs

この状態 (C_{bodyDG} << 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 電極に直接触れた場合、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} << C_{fs} なので、C_{bodyDG} を削除できます。この場合の等価回路は図 187 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。

図 187 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。

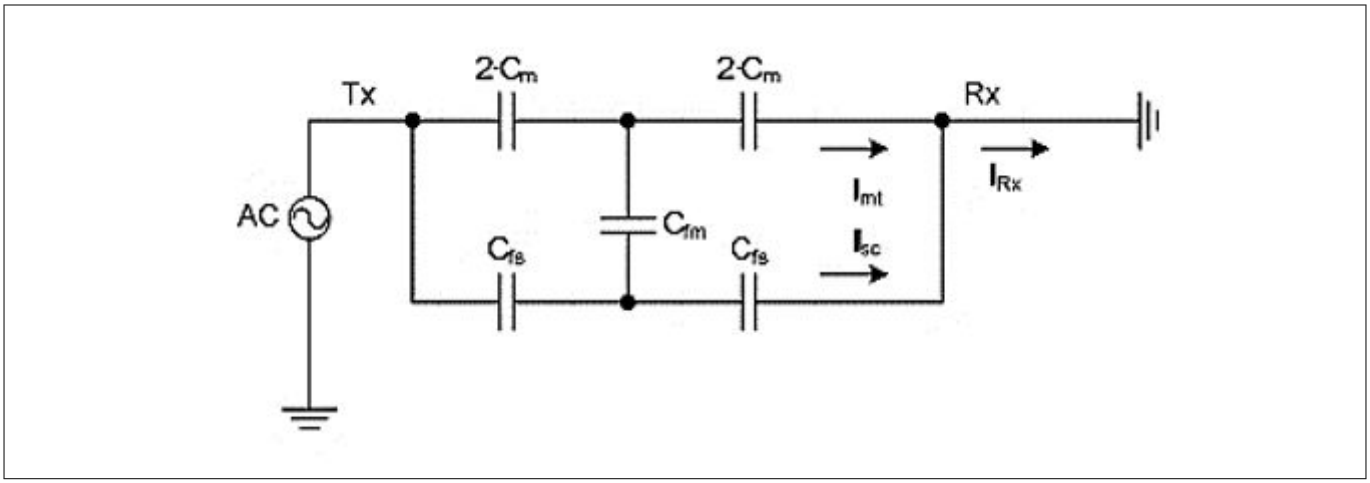


図 187　　Cbody << Cfs の場合の CSX センサーの等価回路

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} << 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} << 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 方式で形成される等価静電容量を図 188 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。シグマ デルタ変換器から、raw カウントは 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。

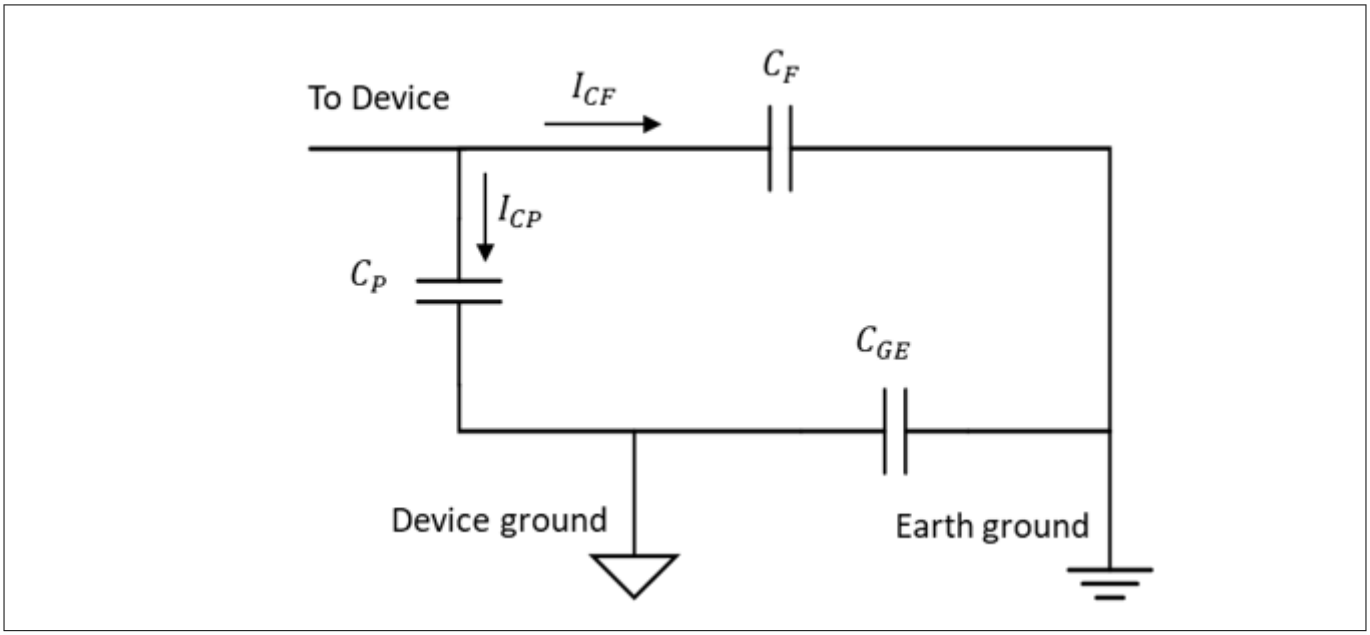


図 188 CSD センサーの等価回路

$$I = I_{CP} + I_{CF}$$

式 82 CSD 方式で IDAC から引き出される電流

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、デバイスのグランドはアースのグラ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、結果の等価回路を図 189 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。

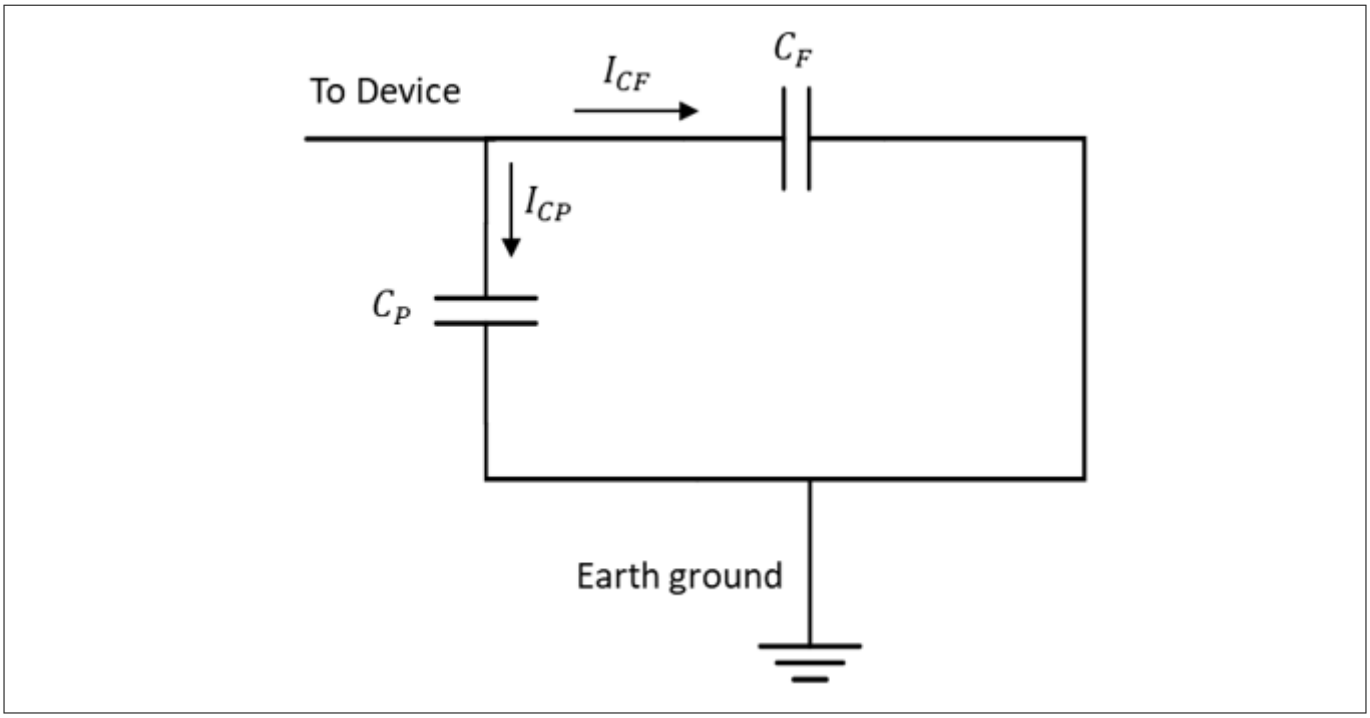


図 189　　主電源アプリケーション用の CSD センサーの等価回路

7.6.2.2　バッテリ駆動のアプリケーション

バッテリ駆動のポータブルアプリケーションでは、デバイスのグランドとアースのグラ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図 190 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。

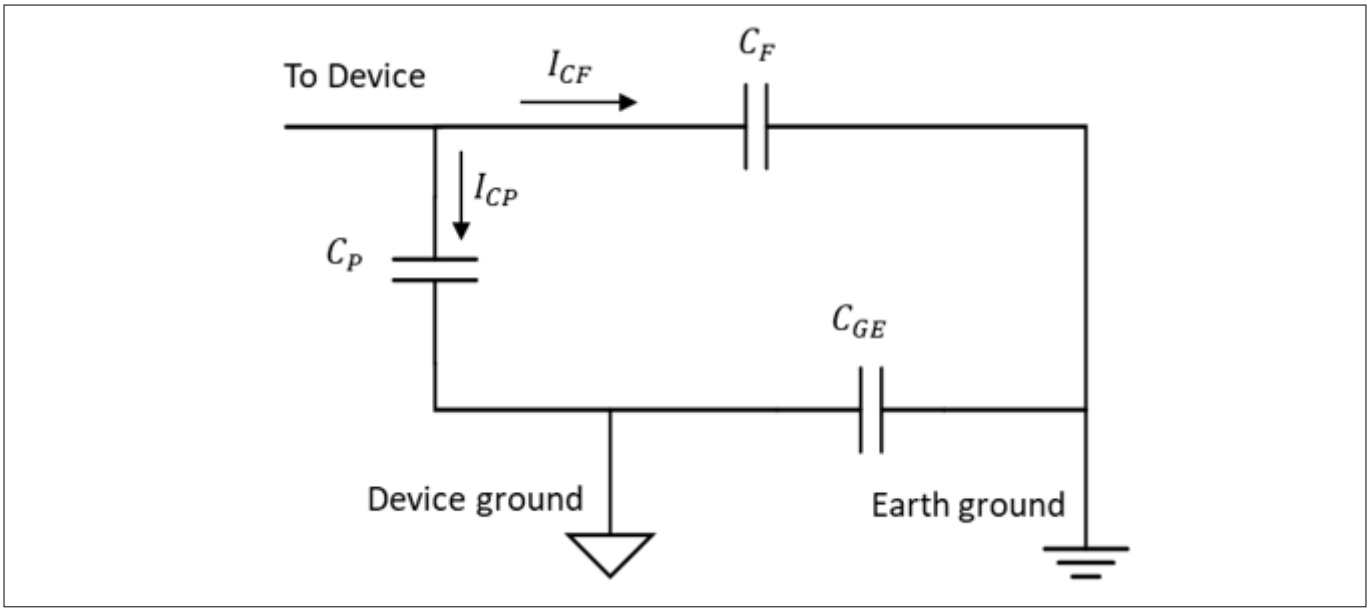


図 190　　バッテリ駆動アプリケーション用の CSD の等価回路

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。

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。

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。図 191 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めます。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。

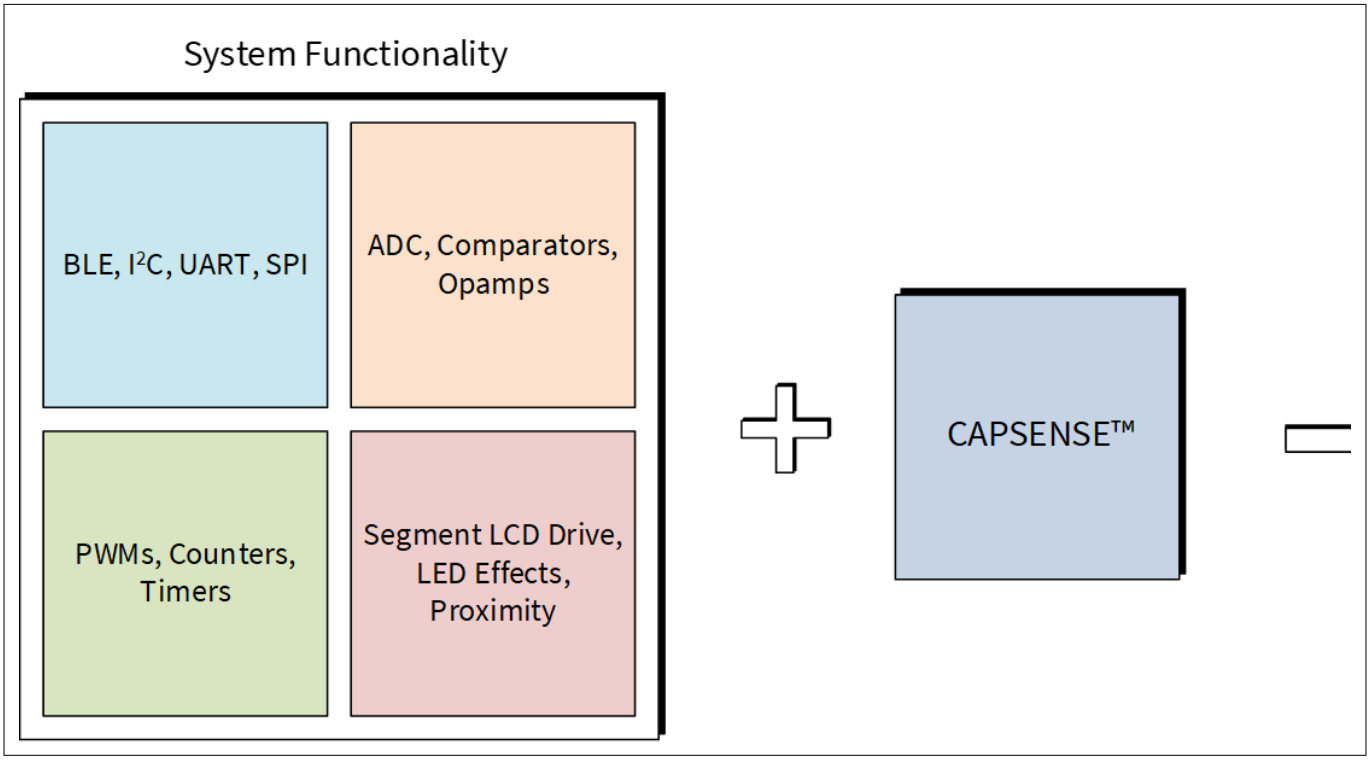


図 191 CAPSENSE™ Plus

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 - PSOC™ 4 入門、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。表 47 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。

8 CAPSENSE™ Plus

表 47 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*)	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 モーター制御	I²C 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	水位監視計	ブザー	I²C 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*** および FOC モーター制御			LED 表示

(続く)

表 47 (続き) 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
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			I²C/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

* PrISM = Precision Illumination Signal Modulation (高精度照度信号変調)

** FOC = Field Oriented Control (フィールド指向制御)

*** BLDC = Brushless DC Motor (ブラシレス DC モーター)

図 192 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。

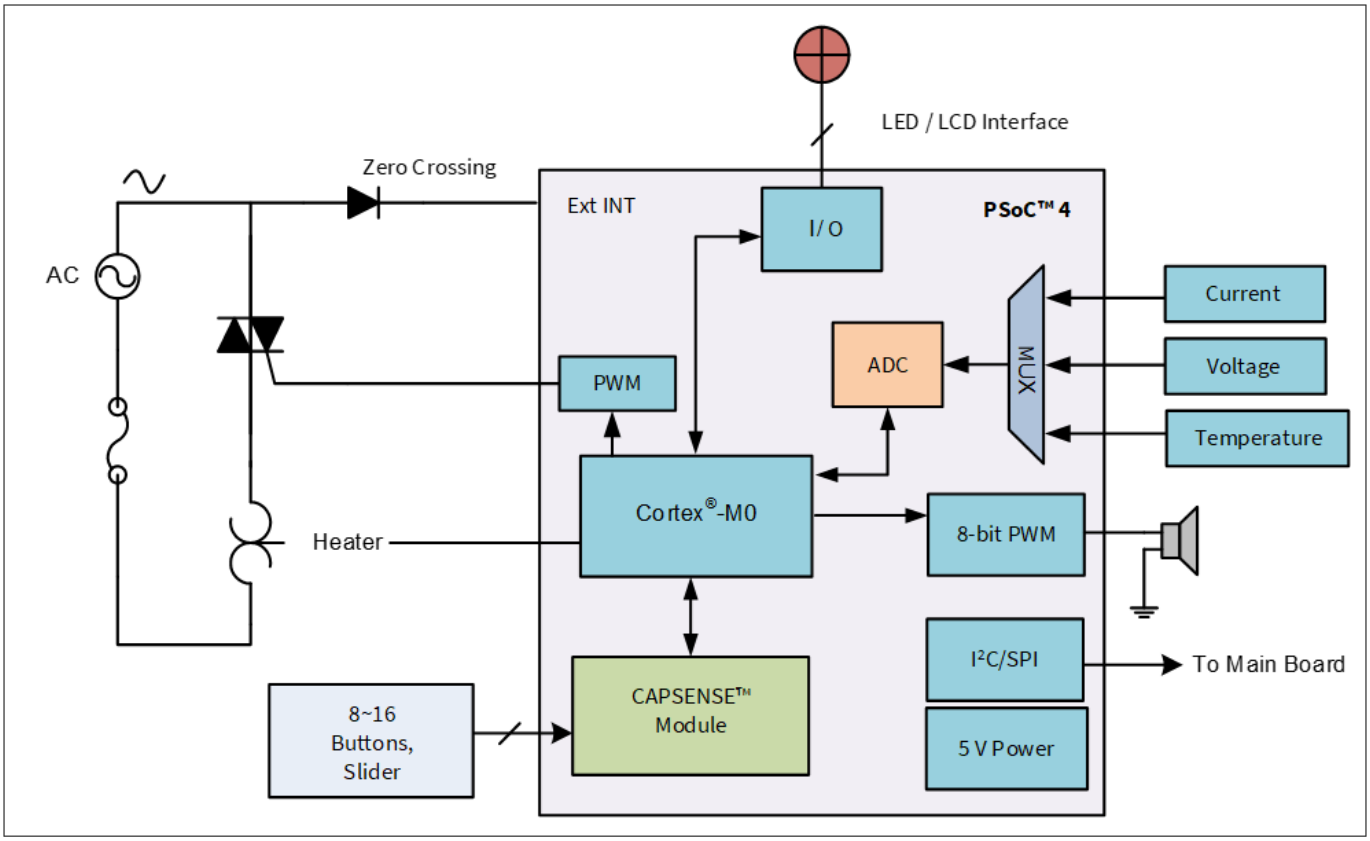


図 192 PSOC™ 4 の CAPSENSE™ Plus システム

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、過電圧および高温度状態を検出します。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ーはユーザー 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。

8 CAPSENSE™ Plus

図 193 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 – PSOC™ Creator での PSOC™ 6 Bluetooth® LE 入門を参照してください。

8 CAPSENSE™ Plus

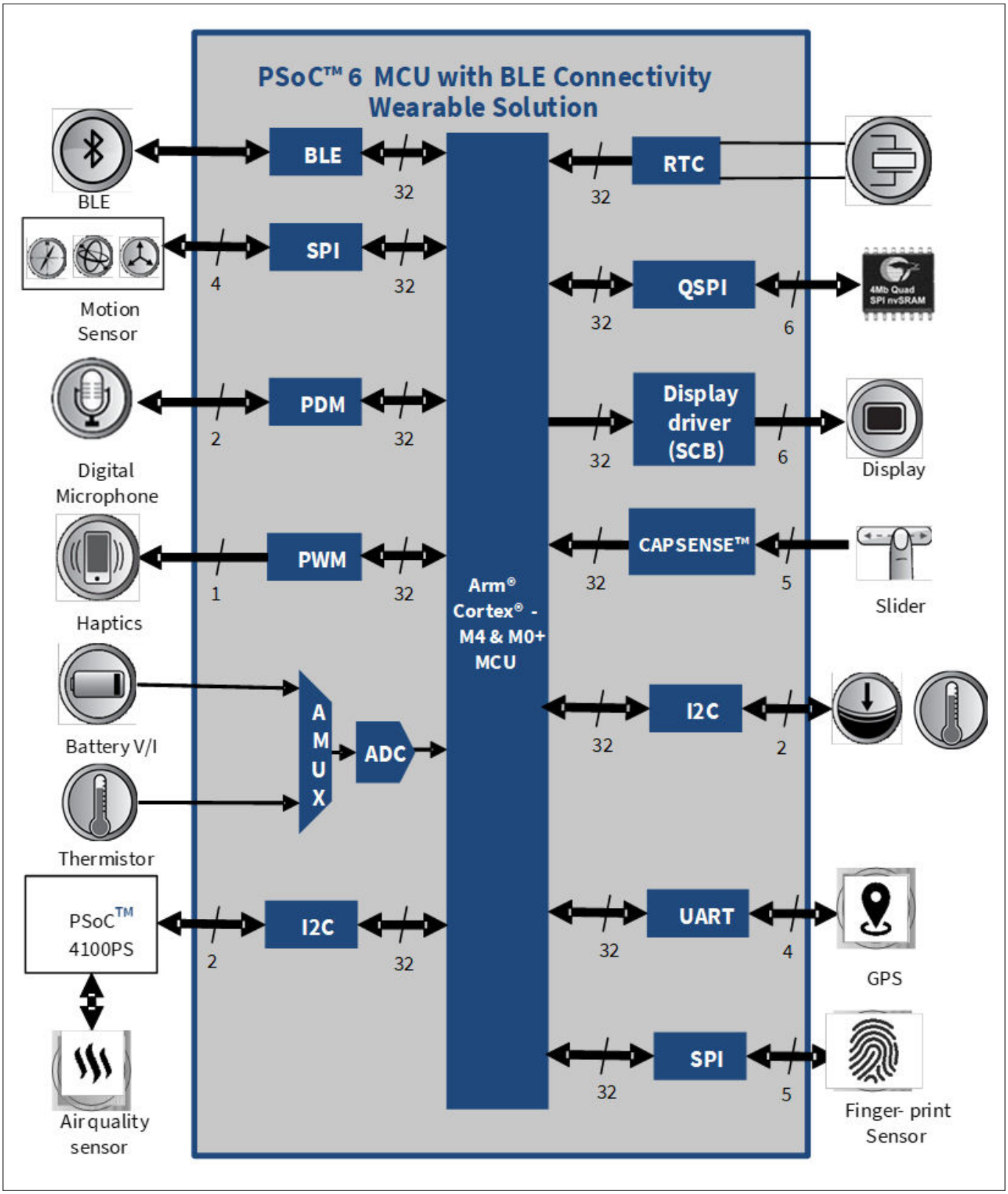


図 193 **Bluetooth® LE 接続搭載の PSOC™ 6 MCU に基づくフィットネストラッカーのアプリケーションのブロック図**

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 の詳細については、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 ドキュメント

- 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

表 8、PSOC™ 4 および PSOC™ 6 CAPSENSE™に対応する Infineon®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われます。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™ software training
- ModusToolbox™ CAPSENSE™ configurator
- ModusToolbox™ CAPSENSE™ tuner

- ModusToolbox™ device configurator
- ModusToolbox™ SmartIO configurator
- PSOC™ Creator to ModusToolbox™
- ModusToolbox™ command line

9.8 アプリケーション ノートおよびデザイン ガイド

® 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 - PSOC™ 4 入門
- AN210781 – PSOC™ Creator での PSOC™ 6 Bluetooth® LE 入門
- 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閲覧したり、CAPSENSE™についてのさまざまなトピックスを検索できます。
- 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。
- インフィニオン開発コミュニティ: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。
- https://www.infineon.com/cms/en/about-infineon/company/contacts/support/: 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、ウェブサイトで登録し、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

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。

CAPSENSE™
Infineon®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 のファミリを含みます。

CAPSENSE™信号
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。

CMOD
変調コンデンサ (CMOD)
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。

CP
寄生容量 (CP)
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。

CSD
CAPSENSE™シグマ デルタ
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。

CSH
シールド タンク コンデンサ (CSH)
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。

MSC
マルチセンスコンバーター (MSC)
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。

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。

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。

SNR
信号対ノイズ比
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。

VREF
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たウィジェットです。

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。

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。

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。

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ってサポートされます。

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要する時間です。

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。

センサー
静電容量センサーを参照してください。

用語集

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。

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。

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。

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。

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。

トラックパッド
タッチパッドを参照してください。

ノイズ
CAPSENSE™ノイズ (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。

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。

用語集

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。

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。

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。

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。

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。

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。

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。

変調器クロック
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 [$(2^N - 1)$/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 。

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。

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。

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。

用語集

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。

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。

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。

自己容量
回路のグランドと電極間の静電容量です。

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。

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。

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。

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。

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。

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。

用語集

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。

電流出力デジタル-アナログ変換器

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。

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。

改訂履歴

発行日	版数	変更内容
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。
-	英文版*A	本版は英語版のみの発行です。英語版の改訂内容: Added dual IDAC support.Updated some schematics in chapter 6.Other minor changes to chapters 3, 5, and 6.
-	英文版*B	本版は英語版のみの発行です。英語版の改訂内容: Added support of CY8C4000 devices.Minor fixes throughout the document.
-	英文版*C	本版は英語版のみの発行です。英語版の改訂内容: Updated the table of device features.Changed IDAC names to sync with new PSOC™ Creator Component terms.Added a schematic checklist.Changed screenshots to match the new Component version.
-	英文版*D	本版は英語版のみの発行です。英語版の改訂内容: Updated Table 1-1 per PSOC™ 4000 datasheet.
-	英文版*E	本版は英語版のみの発行です。英語版の改訂内容: Added firmware design considerations to Chapter 6. Added power supply layout and schematic considerations to Chapter 6. Updated the IMO range for PSOC™ 4000
-	英文版*F	本版は英語版のみの発行です。英語版の改訂内容: Updated to support PSOC™ 4000 and PSOC™ Creator 3.0 SP1.
-	英文版*G	本版は英語版のみの発行です。英語版の改訂内容: Added Reference to Getting started with CAPSENSE™ in **近接 (3 次元)** Renamed 第 2.5 章 to 耐液性 and re-wrote this section. Updated the recommendations for Shield drive that is Csh_tank precharge and C_{MOD} precharge in 第 3.2.7 章 CAPSENSE™ CSD シールド. Added recommendation for setting “API resolution” in Chapter Added guidelines on how to select value of “Sensitivity” parameter in Chapter Updated recommended values of threshold and hysteresis parameters in Chapter 手動チューニングのトレードオフ Added Section Manual Tuning Slider Example. Updated maximum overlay thickness value for sliders in 表 31. Added guideline on maximum thickness for overlays of materials other than acrylic in 第 7.3.2 章 オーバーレイの厚さ. Re-wrote Chapter スライダーの設計. Added recommendations on DC loads in 第 7.4.5 章 Renamed and rewrote 第 7.4.12 章 to 耐液性のレイアウト ガイドライン. Added 第 7.4.13.1 章 外部コンデンサ接続のピン選択. Updated slider related recommendations in Layout rule Updated 電磁適合性 (EMC) に対する考慮事項 , added extensive data on hardware and firmware considerations.
-	英文版*H	本版は英語版のみの発行です。英語版の改訂内容: Added information for the PSOC™ 4 Bluetooth® LE family of devices. Added information for the PRoC Bluetooth® LE family of devices. Updated ground and power layout guidelines in 第 7.4.10 章 and 第 7.4.11 章.

改訂履歴

発行日	版数	変更内容
-	英文版*I	本版は英語版のみの発行です。英語版の改訂内容: Added information for PSOC™ 4200-M family of devices. Added footnote in chapter Slider. Added GPIO source/sink current limit in 表 38. Changed document title to PSOC™ 4 CAPSENSE™ Design Guide – AN85951
-	英文版*J	本版は英語版のみの発行です。英語版の改訂内容: Changed Document Title to “AN85951 – PSOC™ 4 CAPSENSE™ Design Guide” . Updated 設計上の注意事項 Updated ESD 放電の防止. Updated 図 165. Updated リダイレクト. Replaced "Guard Ring" with "Ground Ring".
-	英文版*K	本版は英語版のみの発行です。英語版の改訂内容: Added Table 3-1. Removed chapter 3.2.1 C_{MOD} Precharge. Added chapter CAPSENSE™ in PSOC™ 4xxxM/4xxxL-Series. Updated chapter 配線の経路. Added reference of AN2397. Added recommendation for modulator clock divider in chapter 手動チューニングのトレードオフ. Added 図 162
2015-11-20	*A(英文版*L)	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。英語版の改訂内容: Updated 第 1.1 章. Updated Page Number 図 1. Updated 表 8, 表 36, 表 37, 表 39.
-	英文版*M	本版は英語版のみの発行です。英語版の改訂内容: Updated はじめに. Moved 信号対ノイズ比 (SNR) to 第 2 章. Updated Chapters PSOC™ 4 and PSOC™ 6 MCU and for CAPSENSE™ performance details. Added section to 第 4 章. Added Glossary.
-	英文版*N	本版は英語版のみの発行です。英語版の改訂内容: Added information on mutual-capacitance sensing in PSOC™ 4 device series. Added information on CAPSENSE™ 3.0 changes. Added following sections: • 相互容量センシング • CAPSENSE™ Architecture in PSOC™ 4 S-series Updated following sections: • はじめに • CAPSENSE™ • CAPSENSE™設計および開発ツール • CAPSENSE™性能のチューニング

改訂履歴

発行日	版数	変更内容
2016-05-17	*B(英文版*O)	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。英語版の改訂内容: Added PSOC™ Analog Coprocessor references. Updated 外部コンデンサ接続のピン選択. Updated 開発キット chapter. Updated document title. Updated Copyright notice.
-	英文版*P	本版は英語版のみの発行です。英語版の改訂内容: Updated IDAC sinking mode recommendation. Updated template.
-	英文版*Q	本版は英語版のみの発行です。英語版の改訂内容: Updated Recommended pins for external capacitors.
2017-04-17	*C(英文版*R)	ロゴと著作権を更新しました。
-	英文版*S	本版は英語版のみの発行です。英語版の改訂内容: Added references to PSOC™ 4100S Plus throughout the document. Updated 第 1.2 章 CAPSENSE™の特長 with PSOC™ 4100S Plus features. Updated PSOC™ 4 and PSOC™ 6 CAPSENSE™ development kits with CY8CKIT-149 PSOC™ 4100S Plus prototyping kit. Updated 第 9.8 章 アプリケーション ノートおよびデザイン ガイド with specific list of CAPSENSE™ Application Notes
-	英文版*T	本版は英語版のみの発行です。英語版の改訂内容: Changed document title Added references to PSOC™ 6 MCU features throughout the document Updated CAPSENSE™ generations in PSOC™ 4 and PSOC™ 6 CAPSENSE™ generations in PSOC™ 4 and PSOC™ 6 with generalized architecture block diagram for CSD sensing. Added Gesture in CAPSENSE™ Gesture in CAPSENSE™. Updated , 表 8, and 表 40.
-	英文版*U	本版は英語版のみの発行です。英語版の改訂内容: Added references to PSOC™ 4100PS throughout the document.
-	英文版*V	本版は英語版のみの発行です。英語版の改訂内容: Updated the entire document with references to CY8C62x8 and CY8C62xA devices. Updated the entire document with references to ModusToolbox™. Updated 表 8 with the information of PSOC™ 6 kits. Updated chapter 相互静電容量ボタンの設計 with the information of additional mutual cap key. Removed all references to PRoC Bluetooth® LE devices.
2020-01-15	*D(英文版*W)	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。英語版の改訂内容: Updated SmartSense and 手動チューニング with respect to the latest component. Removed details on different shield drive mode from CAPSENSE™ CSD Updated CAPSENSE™ CSX sensing • Updated figures in PSOC™ Creator • SmartSense, and CAPSENSE™のジェスチャー with respect to the latest component Removed a table in 外部コンデンサ接続のピン選択 chapter Updated 表 3

改訂履歴

発行日	版数	変更内容
-	英文版*X	本版は英語版のみの発行です。英語版の改訂内容: Added Liquid tolerance for Mutual Capacitance Sensing section Removed Mutual Capacitance Button Design section Updated Table 3-2 and Table 3-3 Updated CAPSENSE™ CSX Sensing Method Added ModusToolBox™ section in Chapter 4 Updated SmartSense and Manual Tuning section with respect to the latest component. Updated Slider Tuning Guidelines section Added Tuning Shield Electrode section Updated Gesture chapter with gesture tuning guidelines Updated the Low Power design section Updated Sensor and Device placement section Updated Slider Design section Added Effect of Grounding in CSX method and Effect of Grounding in CSD method section
2021-01-25	*E(英文版*Y)	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。英語版の改訂内容: Updated センサーピンの選択 and chapters.
2021-06-16	*F(英文版*Z)	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。英語版の改訂内容: Added CAPSENSE™ configurator CAPSENSE™ configurator. Moved Tuning Shield Electrode section under 第 5.3.2 章 CSD センシング方式 (第 3 および第 4 世代). Added 第 5.3.7.12 章 CSX ボタンの CM が低いことを確認. Added 第 7.4.3.2 章 相互静電容量ボタンの設計. Added additional layout guidelines in 第 7.4.5 章 センサーとデバイスの配置. Added additional trace routing guidelines in 第 7.4.7 章 配線の経路. Added additional guard trace guidelines in Crosstalk Crosstalk . Added new 第 7.5 章 CAPSENSE™システムのノイズ. Added a single line description on self cap buttons in Self-capacitance button Self-capacitance button . Moved ESD and 電磁適合性 (EMC) に対する考慮事項 under 第 7.5.3 章 外部ノイズ. Moved Effect of grounding on CSX method and effect of grounding on CSD method under 接地の影響.

改訂履歴

発行日	版数	変更内容
2022-03-17	*G(英文版 AA)	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。英語版の改訂内容: Updated to IFX template. Updated CAPSENSE™ features with new Fifth-Generation CAPSENSE™ block and PSOC™ 4100S Max features. Update 表 2. Added new section CAPSENSE™ CSD-RM sensing method (fifth-generation) and CAPSENSE™ CSX-RM sensing method (fifth-generation). Added new section for features 自律スキャン and マルチチャネルの使用. Removed “5.3.2.4 Button widget example” and replaced with ボタン ウィジェット調整. Added new chapter 5.3.2.6 "Touchpad widget tuning". Added new section for Fifth-generation CAPSENSE™ sensing method - CSD-RM センシング方式 (第 5 世代および第 5 世代低消費電力) and CSX-RM センシング方式 (第 5 世代および第 5 世代低消費電力). Updated 表 31, 表 39, and 表 40. Updated Copyright information.
-	英文版 AB	本版は英語版のみの発行です。英語版の改訂内容: Template update
-	英文版 AC	本版は英語版のみの発行です。英語版の改訂内容: In the 表 2, in the Feature column, updated "Noise floor (pk-pk)" to "Noise floor (rms)".
-	英文版 AD	本版は英語版のみの発行です。英語版の改訂内容: Updated link references Deleted CSD Finger Detection Criteria Deleted CSX Finger Detection Criteria Added 同期スキャン section Added CIC2 section Added the section title 第 5 世代および第 5 世代低消費電力の利点
-	英文版 AE	• 本版は英語版のみの発行です。英語版の改訂内容: Updated CIC2 section • Updated SmartSense を使用したハードウェア パラメーターの決定 section • Updated 表 10 • Updated 表 16 • Updated センス クロック ソース
-	英文版 AF	本版は英語版のみの発行です。英語版の改訂内容: Fixed broken links. Removed " 5.3.2.6 Touchpad widget tuning". Removed " 5.3.3.5 Touchpad widget tuning". Removed " 5.3.5.7 Touchpad widget tuning".
2024-10-03	*H (英文版 AG)	これは英語版 001-85951 Rev. AG を翻訳した日本語版 001-89548 Rev. *H です。英語版の改訂内容: Information about CIC2 is updated in Chapter 3. Updated measuring method for sense clock divider and added information about Cfine and CDAC configuration in Chapter 5. Updated LDO recommendation information in VDDA ノイズを低減するための推奨事項 section.

商標

The Bluetooth® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc., and any use of such marks by Infineon is under license.

Trademarks

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

Edition 2024-10-03

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2024 Infineon Technologies AG

All Rights Reserved.

Do you have a question about any aspect of this document?

Email: erratum@infineon.com

Document reference
IFX-jaz1649247577419

重要事項

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

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

警告事項

技術的要件に伴い、製品には危険物質が含まれる可能性があります。当該種別の詳細については、インフィニオンの最寄りの営業所までお問い合わせください。

インフィニオンの正式代表者が署名した書面を通じ、インフィニオンによる明示の承認が存在する場合を除き、インフィニオンの製品は、当該製品の障害またはその使用に関する一切の結果が、合理的に人的傷害を招く恐れのある一切の用途に使用することはできないこと予めご了承ください。