AN66269

CY8C20x34 CapSense®デザイン ガイド

文書番号: 001-79091 Rev. *B

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
http://www.cypress.com

Copyright

© Cypress Semiconductor Corporation, 2010-2019. 本書面は，Cypress Semiconductor Corporation 及び Spansion LLC を含むその子会社（以下「Cypress」という。）に帰属する財産である。本書面（本書面に含まれ又は言及されているあらゆるソフトウェア若しくはファームウェア（以下「本ソフトウェア」という。）を含む）は，アメリカ合衆国及び世界のその他の国における知的財産法令及び条約に基づき Cypress が所有する。Cypress はこれらの法令及び条約に基づく全ての権利を留保し，本段落で特に記載されているものを除き，その特許権，著作権，商標権又はその他の知的財産権のライセンスを一切許諾しない。本ソフトウェアにライセンス契約書が伴っておらず，かつ Cypress との間で別途本ソフトウェアの使用方法を定める書面による合意がない場合，Cypress は，(1) 本ソフトウェアの著作権に基づき，(a) ソースコード形式で提供されている本ソフトウェアについて，Cypress ハードウェア製品と共に用いるためにのみ，かつ組織内部でのみ，本ソフトウェアの修正及び複製を行うこと，並びに (b) Cypress のハードウェア製品ユニットに用いるためにのみ，(直接又は再販売者及び販売代理店を介して間接のいずれかで) 本ソフトウェアをバイナリーコード形式で外部エンドユーザーに配布すること，並びに (2) 本ソフトウェア (Cypress により提供され，修正がなされていないもの) が抵触する Cypress の特許権のクレームに基づき，Cypress ハードウェア製品と共に用いるためにのみ，本ソフトウェアの作成，利用，配布及び輸入を行うことについての非独占的で譲渡不能な一身専属的ライセンス（サブライセンスの権利を除く）を付与する。本ソフトウェアのその他の使用，複製，修正，変換又はコンパイルを禁止する。

適用される法律により許される範囲内で，Cypress は，本書面又はいかなる本ソフトウェア若しくはこれに伴うハードウェアに関しても，明示又は黙示をとわず，いかなる保証（商品性及び特定の目的への適合性の黙示の保証を含むがこれらに限られない）も行わない。いかなるコンピューティングデバイスも絶対に安全ということはない。従って，Cypress のハードウェアまたはソフトウェア製品に講じられたセキュリティ対策にもかかわらず，Cypress は，Cypress 製品への権限のないアクセスまたは使用といったセキュリティ違反から生じる一切の責任を負わない。加えて，本書面に記載された製品には，エラッタと呼ばれる設計上の欠陥またはエラーが含まれている可能性があり，公表された仕様とは異なる動作をする場合がある。適用される法律により許される範囲内で，Cypress は，別途通知することなく，本書面を変更する権利を留保する。Cypress は，本書面に記載のある，いかなる製品若しくは回路の適用又は使用から生じる一切の責任を負わない。本書面で提供されたあらゆる情報（あらゆるサンプルデザイン情報又はプログラムコードを含む）は，参照目的のためのみに提供されたものである。この情報で構成するあらゆるアプリケーション及びその結果としてのあらゆる製品の機能性及び安全性を適切に設計，プログラム，かつテストすることは，本書面のユーザーの責任において行われるものとする。Cypress 製品は，兵器，兵器システム，原子力施設，生命維持装置若しくは生命維持システム，蘇生用の設備及び外科的移植を含むその他の医療機器若しくは医療システム，汚染管理若しくは有害物質管理の運用のために設計され若しくは意図されたシステムの重要な構成部分としての使用，又は装置若しくはシステムの不具合が人身傷害，死亡若しくは物的損害を生じさせるようなその他の使用（以下「本目的外使用」という。）のためには設計，意図又は承認されていない。重要な構成部分とは，それの不具合が装置若しくはシステムの不具合を生じさせるか又はその安全性若しくは実効性に影響すると合理的に予想できるような装置若しくはシステムのあらゆる構成部分をいう。Cypress 製品のあらゆる本目的外使用から生じ，若しくは本目的外使用に関連するいかなる請求，損害又はその他の責任についても，Cypress はその全部又は一部をとわず一切の責任を負わず，かつ Cypress はそれら一切から本書により免除される。Cypress は Cypress 製品の本目的外使用から生じ又は本目的外使用に関連するあらゆる請求，費用，損害及びその他の責任（人身傷害又は死亡に基づく請求を含む）から免責補償される。

Cypress，Cypress のロゴ，Spansion，Spansion のロゴ及びこれらの組み合わせ，WICED，PSoC，CapSense，EZ-USB，F-RAM，及び Traveo は，米国及びその他の国における Cypress の商標又は登録商標である。Cypress のより完全な商標のリストは，cypress.com を参照すること。その他の名称及びブランドは，それぞれの権利者の財産として権利主張がなされている可能性がある。

目次

1. はじめに 5

1.1 概要 5
1.2 サイプレス CapSense の文書体系 5
1.3 CY8C20x34 CapSense ファミリ機能 7
1.4 本書の表記法 8

2. CapSense 技術 9

2.1 CapSense の原理 9
2.2 CY8C20x34 の CapSense メソッド 11

3. CapSense 設計ツール 13

3.1 概論 13
3.2 ユーザー モジュール概要 15
3.3 CapSense ユーザー モジュール グローバル アレイ 15
3.4 CSA_EMC ユーザー モジュールのパラメーター 17

4. ユーザー モジュールによる CapSense 性能のチューニング 21

4.1 一般的な注意事項 21
4.2 CSA_EMC ユーザー モジュール – チューニング ガイド 22
4.3 推奨 C_{INT} 値 23
4.4 センサーC_Pの測定 23
4.5 CSA_EMC クロックの予測 24
4.6 整定時間の設定 24
4.7 CapSense データの監視 25
4.8 SNR の向上方法 25
4.9 チューニングの例 25

5. 設計の注意事項 32

5.1 オーバーレイの選択 32
5.2 ESD 保護 33
5.3 電磁環境適合性 (EMC) の注意点 34
5.4 ソフトウェアのフィルター処理 35
5.5 消費電力 36
5.6 ピンの割り当て 37
5.7 プリント基板レイアウトのガイドライン 38

6. リソース 39
6.1 ウェブサイト 39
6.2 データシート 39
6.3 テクニカル リファレンス マニュアル 39
6.4 開発キット 39
6.5 サンプル ボード ファイル 40
6.6 PSoC Designer 42
6.7 PSoC Programmer 42
6.8 CapSense データ表示ツール 42
6.9 コード例 43
6.10 デザイン サポート 43
用語集 44
改訂履歴 50

1. はじめに

1.1 概要

本書では、容量タッチセンシング (CapSense®) 機能を、CapSense コントローラーの CY8C20x34 ファミリに実装するための設計ガイドを説明します。本ガイドでは以下の項目について説明します。

- CapSense デバイスの CY8C20x34 ファミリの機能
- CapSense の動作原理
- CapSense デザイン ツールの概要
- 最適な性能を得るための CapSense タッチセンシング システムの詳細チューニングガイド
- 段階的な CapSense チューニング例
- CapSense に対する電気および機械システム設計の検討
- CapSense をシステムに組み込むための追加リソースおよびサポート

1.2 サイプレス CapSense の文書体系

図 1-1 と表 1-1 でサイプレス CapSense 文書体系をまとめます。これらのリソースにより、CapSense 製品設計をする際に必要な情報に迅速にアクセスできます。図 1-1 は、容量センシングの製品設計サイクルの代表的なフローを示したものです。本ガイドの情報は、緑色でハイライト表示したトピックに最も関連しています。表 1-1 には、図 1-1 で付番された各タスクをサポートする文書へのリンクが記載されています。

図 1-1. 設計サイクル フロー

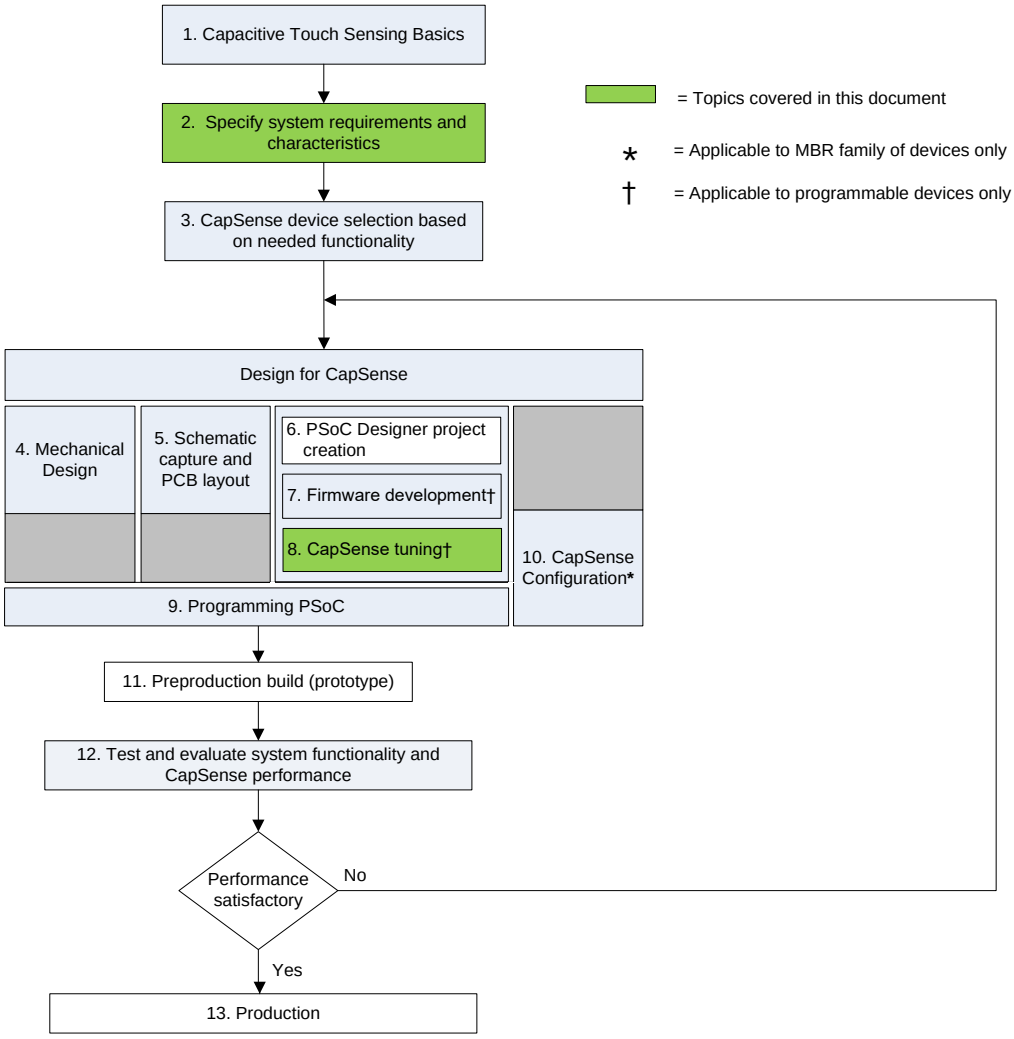


表 1-1.図 1–1 中に番号付けされた設計タスクをサポートしているサイプレスの文書

図 1–1 内の設計タスク番号	関連する Cypress CapSense®文書
1	■ CapSense®入門
2	■ CapSense®入門 ■ CY8C20x34 CapSense® デバイスデータシート ■ PSoC ファミリ向け CapSense® 設計ガイド (本文書)
3	■ CapSense®入門
4	■ CapSense®入門
5	■ CapSense®入門
6	■ PSoC Designer ユーザー ガイド

図 1-1 内の設計タスク番号	関連する Cypress CapSense®文書
7	■ アセンブリ言語ユーザー ガイド ■ C 言語コンパイラ ユーザー ガイド ■ CapSense コードの例 ■ PSoC ファミリ向け技術リファレンス マニュアル ((CY8C20x34 用)
8	■ PSoC ファミリ向け CapSense®デザイン ガイド (本文書) ■ PSoC ファミリ向け CapSense® ユーザー モジュール データシート (CSA_EMC) ■ 「CapSense Data Viewing Tools」 - AN2397 ■ 「CapSense Controller Code Examples」デザイン ガイド
9	■ プログラマ ユーザー ガイド ■ MiniProg3 ユーザー ガイド
11	■ CapSense コードの例

1.3 CY8C20x34 CapSense ファミリ機能

サイプレスの CY8C20x34 は、低電力、高性能、プログラム可能な CapSense コントローラー ファミリで、以下の機能があります。

1.3.1 高度なタッチ センシング機能

- プログラム可能な静電容量センシング要素
  - CapSense ボタン、スライダー、近接センサーと CSA_EMC 容量センシング技術の組み合せをサポート
  - ボタンとスライダーを実装する集積 API を含む
  - 最大 25 の容量センサーと 6 つのスライダーをサポート
  - CY8C20x24 は 25 の容量センサーと 1 つのスライダーをサポート
  - 最大 2 cm の近接センシングをサポート (プリント基板トレース搭載)
- 高いノイズ耐性
  - CSA_EMC は、厳しい伝導性および放射性ノイズがある状態で使用する際にも優れたノイズ耐性を発揮します。
- 低消費電力
  - 電力消費を最適化するための 2 つの電源モード
  - 1.5mA アクティブ モード電流、2.6µA スリープ モード電流

1.3.2 デバイスの特長

- 高性能かつ低消費電力 M8C ハーバード アーキテクチャ プロセッサ
  - 最大 2 MIPS、12MHz 内部クロック、外部クロック信号
- 柔軟性のあるオンチップ メモリ
  - 最大 8KB のフラッシュと 512B の SRAM
  - エミュレート EEPROM をサポート
- プログラム可能な高精度クロック

- ☐ 内部主振動子 (IMO): 6/12MHz ± 5%
- ☐ ウォッチドッグおよびスリープ タイマー用の 32kHz 内部低速振動子

■ 拡張された汎用入出力 (GPIO) の特長

- ☐ プログラム可能なピン コンフィギュレーションを備えた最大 28 の汎用 I/O (GPIO)
- ☐ すべての GPIO で 20mA のシンク電流に対応
- ☐ すべての GPIO で、耐環境性に優れた内部プルアップ、HI-Z、オープンドレイン、ストロングの駆動モード

■ ペリフェラルの特長

- ☐ 13 ビットのプログラム可能タイマー
- ☐ ウォッチドッグ タイマーとスリープ タイマー
- ☐ I^2C マスター – 100kHz のクロック、I^2C スレーブ– 最大 400kHz
- ☐ SPI マスターおよびスレーブ – 設定可能範囲 46.9kHz ～ 3MHz

■ 動作条件

- ☐ 動作電圧: 2.4V～5.25V
- ☐ 温度範囲: –40°C～+85°C

1.4 本書の表記法

表記法	使用法
`Courier New` フォント	ファイルの場所、ユーザーが入力したテキスト、ソース コードを示す: `C:\ ...cd\icc\`
*イタリック*フォント	ファイル名および参考ドキュメントを示す: 「*PSoC Designer User Guide*」にある *sourcefile.hex* ファイルを参照
[角括弧、太字]	操作手順でキーボード コマンドを示す: [**Enter**]または[**Ctrl**] [**C**]
File > Open	メニュー パスを示す **File > Open > New Project**
太字	操作手順でコマンド、メニュー パス、アイコン名を示す: **File** アイコンをクリックして、**Open** をクリックする
Times New Roman フォント	数式を示す: 2 + 2 = 4
灰色のボックス内のテキスト	製品の注意点や製品固有の機能を示す

2. CapSense 技術

2.1 CapSense の原理

CapSense はタッチ センサー技術で、センサーである CapSense コントローラー上の各 I/O pin の静電容量を計測することによって機能します。図 2-1 に示すように、n 個のセンサーを備えたデザインにおいて、各センサー ピンの総容量を $C_{X,1}$～$C_{X,n}$ の値を持つ等価集中コンデンサとしてモデル化することができます。CY8C20x34 デバイスへの内部回路により、各 C_X の大きさが後処理用に格納されているデジタル コードに変換されます。もう一つのコンポーネント、集積コンデンサ C_{INT} は、CapSense コントローラーの内部回路で使用されます。

図 2-1 CY8C20x34 PSoC デバイスへの CapSense 固有デバイスの実装

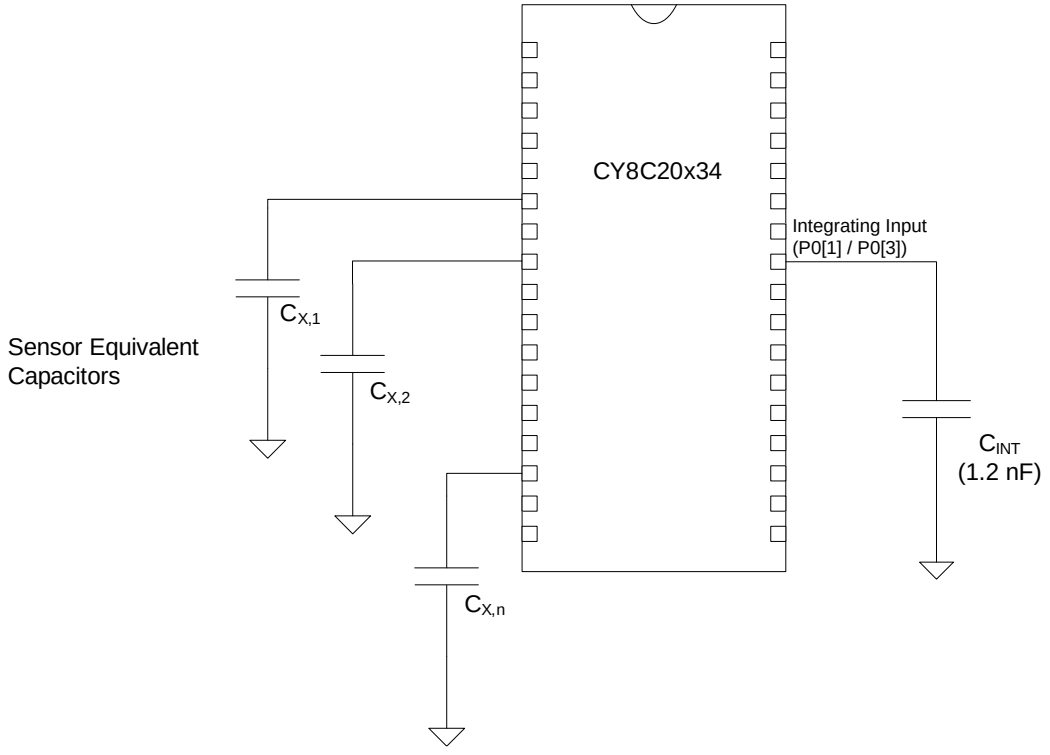


図 2-1 に示すように、センサーの各 I/O ピンは必要に応じて配線、ビアまたはその両方でセンサー パッドに接続されます。オーバーレイはセンサーパッドを覆う非導電のカバーで、製品のタッチインターフェースを構成します。指がオーバーレイに接触すると、人体の伝導性と大きさにより、グランドに接続された導体板がセンサー パッドと平行に置かれるのと同じ状況になります。これを図 2-2 に示します。この配置は平行板コンデンサを構成し、その静電容量は以下のようになります:

$$C_F = \frac{\varepsilon_0 \varepsilon_r A}{D} \qquad \text{式 1}$$

ここではそれぞれ次を示します:

C_F = センサーを覆うオーバーレイに接触する指により生じた静電容量

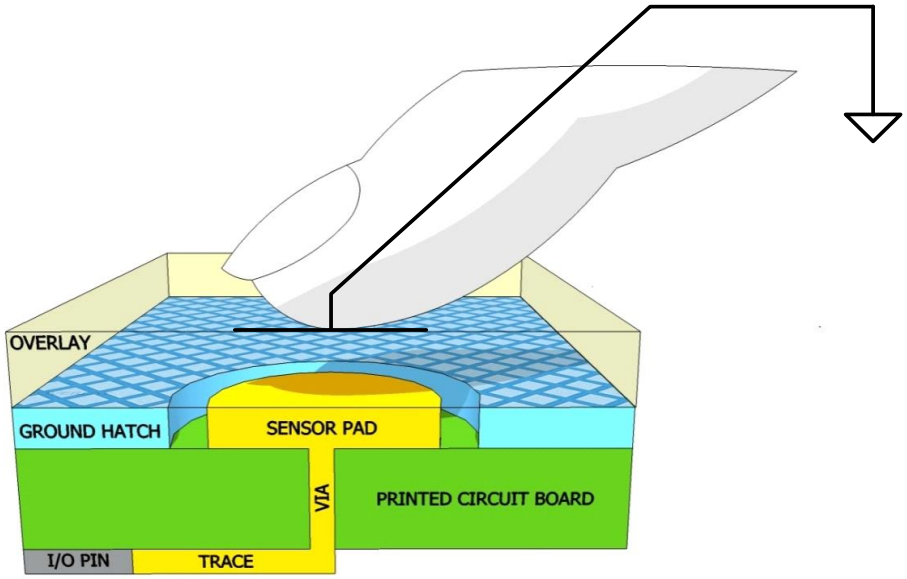
ε_0 = 真空の誘電率

ε_r = オーバーレイの比誘電率

A = 指とセンサー パッドが重なっている面積

D = オーバーレイの厚さ

図 2-2. 指でセンサーが起動した時の典型的な CapSense 基板の断面図



平行板コンデンサに加えて、オーバーレイに接触している指は、それ自身とすぐ近くにある他の導体との間に静電結合を引き起こします。このフリンジ電界の影響は平行板コンデンサと比較して通常小さく、無視することができます。

指がオーバーレイに触れなくても、センサーI/O ピンは寄生容量 (C_P) があります。C_P は、CapSense コントローラー内部の寄生容量および、センサバッド、配線、およびビア、ならびに グランド面、他の配線、製品のシャーシまたは同封物に含まれる金属の間で結合した電場などとの、組み合わせの結果として生じます。CapSense コントローラーは、センサー ピンに接続しているすべての静電容量 (C_X) を計測します。

指がセンサーに触れていない場合:

$$C_X = C_P \qquad \text{式 2}$$

指がセンサーパッド上にある場合は、C_X は C_P と C_F の和に等しい:

$$C_X = C_P + C_F \qquad \text{式 3}$$

一般的に、C_P は C_F より桁違いに大きい値です。通常、C_P は 6pF～15pF の範囲ですが、極端な場合は、50pF になることもあります。C_F は通常 0.1pF から 0.4pF の範囲です。C_P の大きさは、CapSense システムのチューニング時にはきわめて重要であり、これについては「ユーザー モジュールによる CapSense 性能のチューニング」で説明されています。

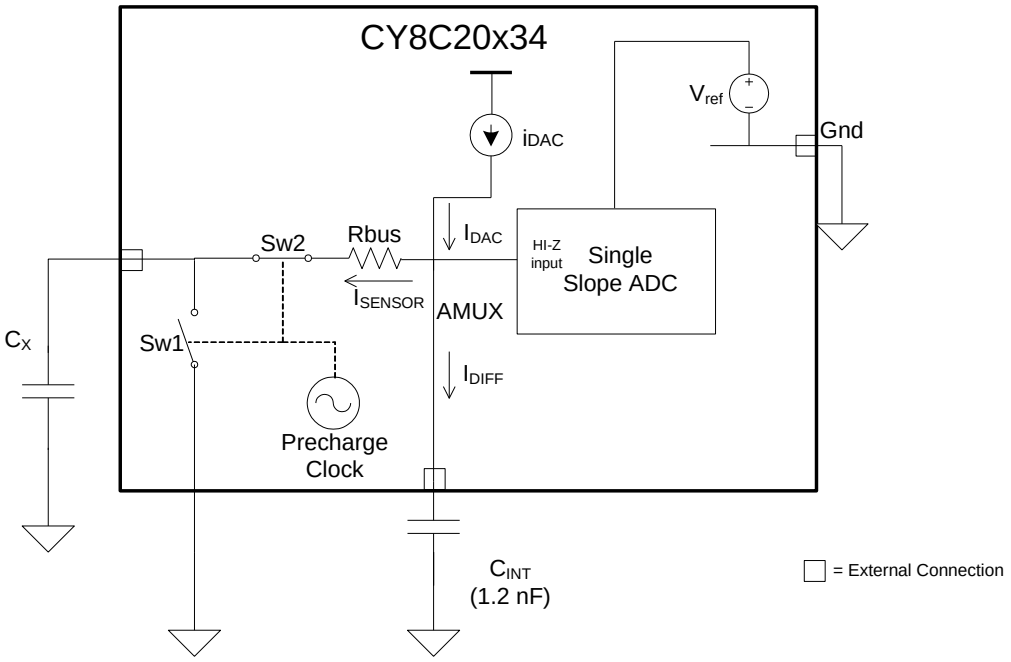
2.2 CY8C20x34 の CapSense メソッド

センサー容量 (C_X) をデジタル コードに変換するために、CapSense メソッド、CapSense 逐次比較型電磁環境適合性 (CSA_EMC)、が CY8C20x34 デバイスでサポートされています。CSA_EMC メソッドは、PSoC 設計者ユーザー モジュールの形で実装され、次のセクションで説明します。

2.2.1 CapSense 逐次比較型電磁環境適合性 (CSA_EMC)

CY8C20x34 デバイスで使用される CSA_EMC メソッドでは、図 2-3 に示すように、C_X がスイッチドキャパシター回路に組み込まれます。

図 2-3. CSA_EMC ブロック図



定電流源 (iDAC) は、I_{DAC} の電流をアナログ MUX に提供します。スイッチ Sw1 と Sw2 によって、順にアナログ MUX (AMUX) バスとグランドに交互に接続されるセンサー (C_X) は、AMUX バスから I_{SENSOR} の電流を使用します。I_{SENSOR} は C_X の大きさに正比例します。スイッチ Sw1 と Sw2 は、プリチャージ クロックと呼ばれる非オーバーラップ クロックによってクロック入力されます。

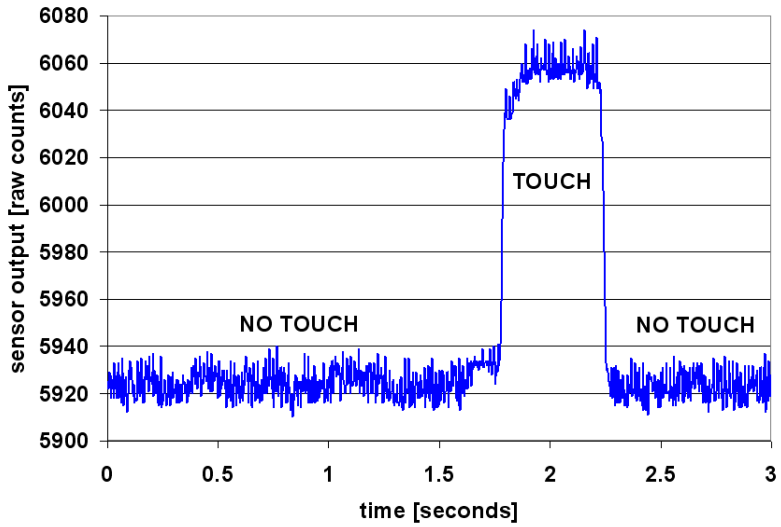
積分コンデンサ C_{INT} は、差分電流 I_{DIFF} (I_{DAC} と I_{SENSOR} の差分)を積分し、その電位.を上げます。このチャージインテグレーションは、C_{INT} での電圧上昇が平衡に達して I_{SENSOR} が IDAC に等しくなるまで続きます。この積分時間を整定時間と言います。

シングルスロープ ADC を使用して、C_{INT} の平衡電位を、raw カウントという C_X に比例するデジタル出力カウントに変換します。この raw カウントは、高レベルのアルゴリズムによって解釈され、センサーの状態を判断します。

I_{DAC} 電流は、C_{INT} の平衡電圧が ADC の線形変換領域内になるようにする逐次比較方式を使用して設定されます。

図 2-4 は、センサーに指でタッチしてから離すまでの多くの連続スキャンの CSA_EMC raw カウントをプロットしたものです。「CapSense の原理」で説明したように、指の接触により C_X が C_F 分増加し、その結果、raw カウントが比例して増加します。定常状態の raw カウント レベルの動きと事前設定された閾値を比較することで、センサーがオン (タッチ) 状態であるかオフ (非タッチ) 状態であるかを高レベル アルゴリズムにより決定できます。

図 2-4. 指でタッチ中の CSA_EMC raw カウント



CSA_EMC CapSense アルゴリズム は、RF 妨害がある場合でも正常に 機能するように設計されています。CSA_EMC は、CapSense が導電干渉、AC ノイズ、およびインバーター、変圧器、電源などその他のノイズ源に曝されるアプリケーションで使用します。これについては、「低レベル パラメーター」で詳しく説明します。

3. CapSense 設計ツール

3.1 概論

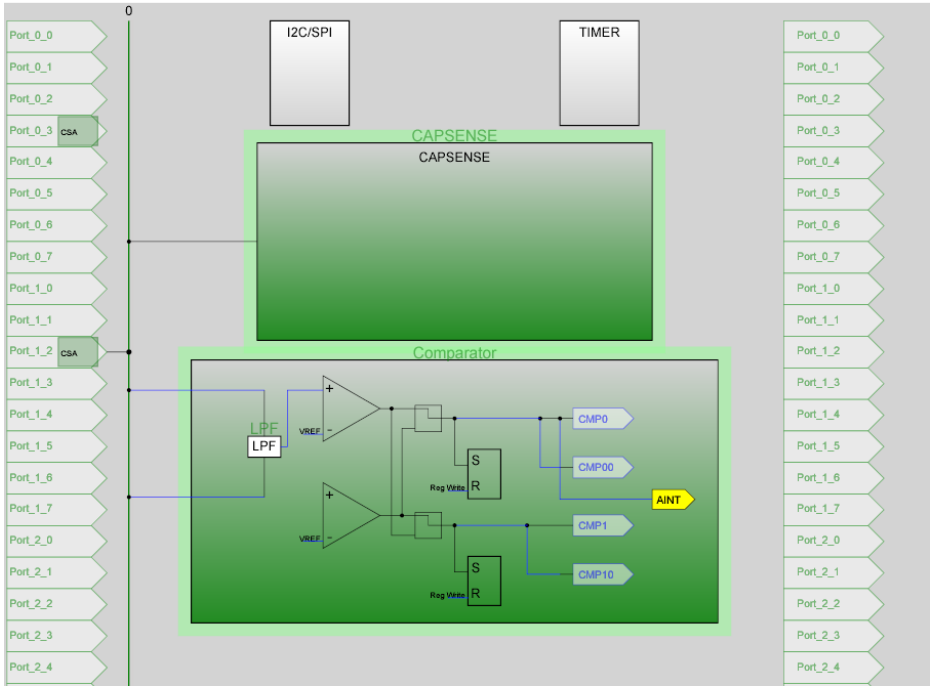
サイプレスは、CapSense 容量タッチ センシング アプリケーション開発用のハードウェアおよびソフトウェア ツール一式を取り揃えています。CY8C20x34 ファミリの基本開発システムには、本章で説明するコンポーネントが含まれます。注文情報については、「リソース」を参照してください。

3.1.1 PSoC Designer およびユーザー モジュール

Cypress の専用統合設計環境である PSoC Designer では、アナログおよびデジタル ブロックの構成、ファームウェアの開発、設計のチューニングとデバッグが可能です。アプリケーションは、ユーザー モジュール のライブラリを使用するドラッグアンドドロップ方式の設計環境で開発されます。ユーザー モジュールは、デバイス エディターGUI、あるいはファームウェアで特定のレジスタに書込むことで設定されます。PSoC Designer は、専用 C コンパイラおよび組み込みのプログラマと一体化しています。複雑なデザイン用には pro コンパイラがあります。

CSA_EMC ユーザー モジュールは、スイッチトキャパシタ回路、アナログ マルチプレクサ、コンパレータ、デジタル カウント機能、高レベル ソフトウェア ルーチン (API) を使用して容量タッチ センサーを実装します。その他のアナログおよびデジタル周辺機器用のユーザー モジュールを使用し、I²C、SPI、TX8、およびタイマーなどの追加機能を実装することができます。

図 3-1. PSoC Designer デバイス エディター



3.1.1.1 CapSense ユーザー モジュール入門

PSoC Designer に新しい CY8C20x34 プロジェクトを作成する場合は、以下の手順に従います。

1. ターゲット デバイスとして、CY8C20x34 を選択し、新しい PSoC Designer プロジェクトを作成します。
2. 専用ピン (I²C または LCD など) が必要なユーザー モジュールを選択、配置します。ポートとピンを割り当てます。
3. CSA_EMC ユーザー モジュールを選択、配置します。
4. CSA_EMC ユーザー モジュールを右クリックし、CSA_EMC ウィザード にアクセスします。
5. 必要なセンサー数、スライダー数、またはラジアル式スライダー数を設定します。
6. ピン及びユーザー モジュールのグローバル パラメーターを設定します。
7. アプリケーションを生成し、アプリケーションエディターを開きます。
8. ユーザー モジュール データシートからサンプル コードを適合し、ボタン、スライダー、またはタッチ パッドを実装します。

CapSense ユーザー モジュールのコード例については、コード例.を参照してください。

3.1.2 ユニバーサル CapSense コントローラー キット

ユニバーサル CY280-20x34 CapSense コントローラー キットには、プロトタイプ作成とデバッグ処理を簡単にするための制御回路とプラグイン ハードウェアが事前定義されています。MiniProg ハードウェアは、プログラミング用キットに同梱されており、チューニングおよびデータ取得用に I²C- USB ブリッジ ハードウェア (CY3240 – I2C USB ブリッジ) が含まれています。

3.1.3 ユニバーサル CapSense コントローラー モジュール ボード

サイプレスのモジュール基板には、用途に応じたさまざまなセンサー、LED、インターフェースが用意されています。

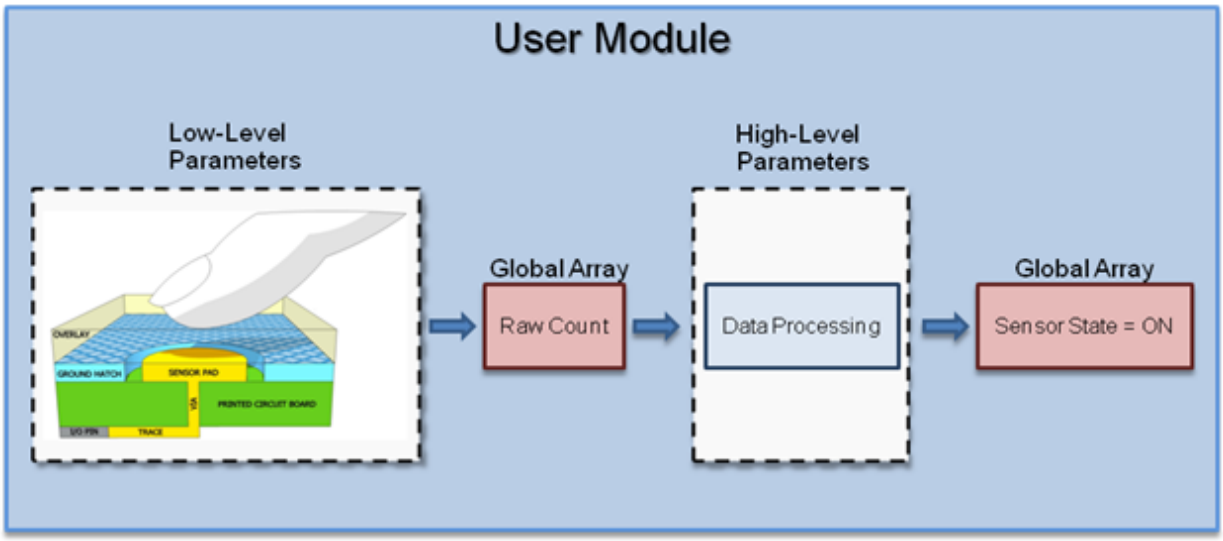
- CY3280-BSM シンプル ボタン モジュール
- CY3280-BMM マトリックス ボタン モジュール
- CY3280-SLM リニア スライダー モジュール
- CY3280-SRM ラジアル スライダー モジュール
- CY3280-BBM ユニバーサル CapSense プロトタイピング モジュール

3.1.4 CapSense データ表示ツール

CapSense の設計中には、チューニングまたはデバッグ作業用に CapSense データ (raw カウント、ベースライン値、差分カウントなど) の監視が必要となる場合が何度もあります。CapSense データ表示ツールには、マルチチャートとブリッジ制御パネルの 2 つがあります。これらのツールについては、アプリケーション ノート「AN2397 – CapSense データ表示ツール」で詳しく説明します。

3.2 ユーザー モジュール概要

図 3-2. ユーザー モジュールのブロック図



ユーザー モジュールには、物理的な検知からデータ処理までの全 CapSense システムが含まれます。ユーザー モジュールの動作は、さまざまなパラメーターを使用して定義します。これらのパラメーターは検知システムのさまざまな部分に影響を及ぼし、グローバル アレイを使用して相互に通信する低レベル パラメーターと高レベル パラメーターに分割できます。

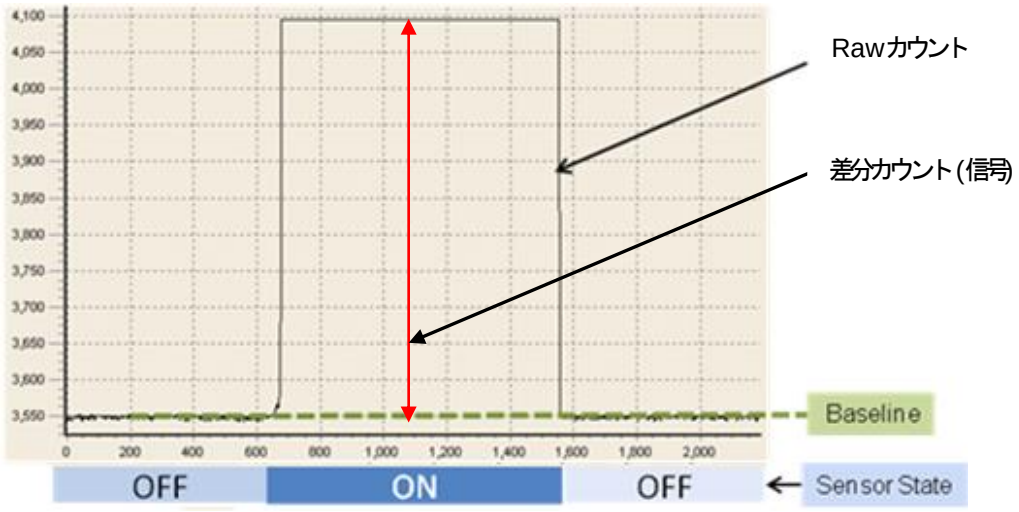
スキャン用センサーのスピードや解像度などの低レベル パラメーターは、物理レイヤで検知方法の動作を定義し、容量から raw カウントへの変換に関連付けます。これらのパラメーターは検知方法の各タイプに対して一意であり、「低レベル パラメーター」で説明します。

デバウンス カウントやノイズ閾値といった高レベル パラメーターでは、raw カウントを処理する方法を定義して、センサーのオン／オフ状態やスライダー上の指の予想位置などの情報を生成します。これらのパラメーターはすべての検知方法で同じであり、「高レベル パラメーター」で説明します。

3.3 CapSense ユーザー モジュール グローバル アレイ

CapSense ユーザー モジュール パラメーターについて習得する前に、CapSense システムで使用する特定のグローバル アレイに精通しておく必要があります。これらのアレイはマニュアルで変更しないでください。ただし、デバッグのプロセスで検査することができます。

図 3-3. raw カウント、ベースライン値、差分カウント、およびセンサー状態



3.3.1 raw カウント

CapSense コントローラーのハードウェア回路でセンサー容量が測定され、結果は、ユーザー モジュール API CSA_EMC_ScanSensor()を呼び出して、raw カウントというデジタル形式で格納されます。

センサーの raw カウントは、そのセンサー容量に比例しています。センサー容量値が増えると、raw カウントも増えます。

CSA_EMC ユーザー モジュールは、センサーの raw カウント値を整数アレイ CSA_EMC_waSnsResult[]に格納します。このアレイは、ヘッダ ファイル CSA_EMC.h で定義されています。

3.3.2 ベースライン

温度や湿度などの段階的な環境変化は、センサーの raw カウントに影響を及ぼし、その結果カウントが変動します。

複雑なベースライン値アルゴリズムは、ユーザー モジュールでこれらの変動を補正する場合に使用します。アルゴリズムでは、ベースライン値変数を使用してこの補正を行います。ベースライン値変数は、raw カウント値の段階的な変動を記録します。つまり、ベースライン値変数は、入力 raw カウント値を入力するデジタルロー パス フィルターの出力を保持します。

ベースライン値アルゴリズムは、ユーザー モジュール API CSA_EMC_UpdateSensorBaseline によって実行されます。

センサーのベースライン値は、整数アレイ CSA_EMC_waSnsBaseline[]に格納されます。このアレイは、ヘッダ ファイル CSA_EMC.h で定義されています。

3.3.3 差分カウント (信号)

センサーの信号とも呼ばれる差分カウントは、センサーの raw カウントとベースライン値間の差分カウントとして定義されています。センサーが非アクティブの場合、差分カウントはゼロです。タッチしてセンサーをアクティブ化すると、差分カウント値はプラスになります。

センサーの差分カウント値は整数アレイ CSA_waSnsDiff[]に格納されます。このアレイは、ヘッダ ファイル CSA_EMC.h で定義されています。

差分カウント変数は、ユーザー モジュール API CSA_EMC_UpdateSensorBaseline()により更新されます。

3.3.4 センサー状態

センサー状態は、物理センサーのアクティブ／非アクティブ状態を表します。指でタッチすると、センサー状態が 0 から 1 に変わり、指を離すと 0 に戻ります。

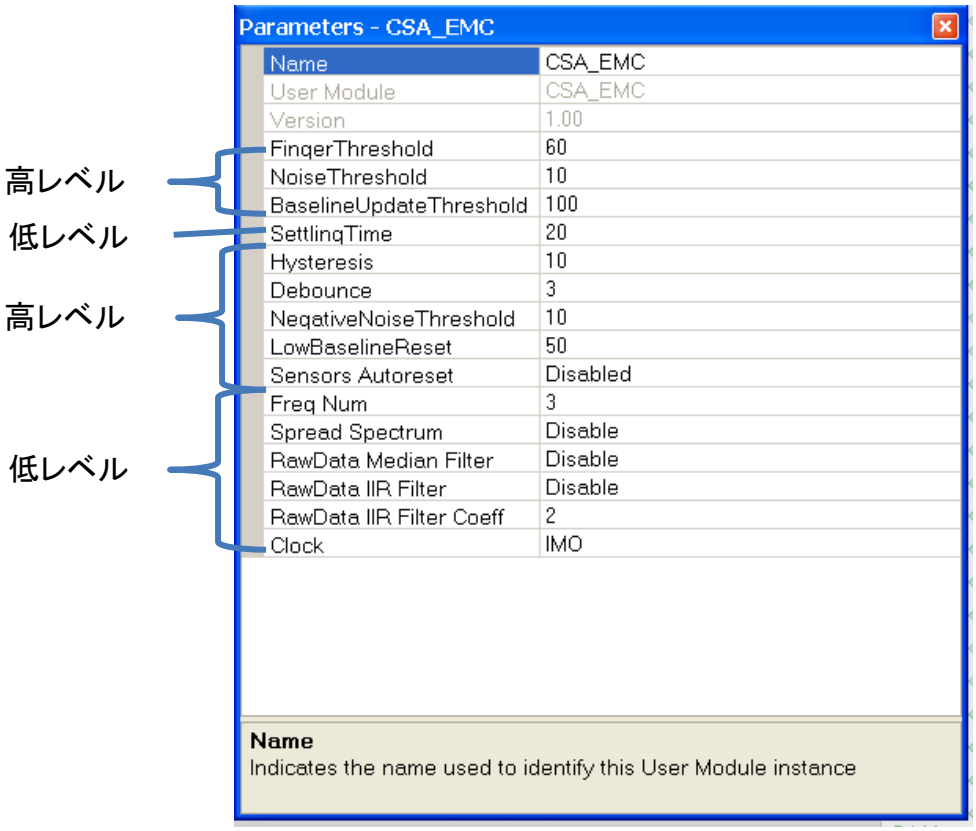
センサー状態は、CSA_EMC_baSnsOnMask[]という名前のバイト アレイに格納されます。各アレイ要素には、8 つの連続センサーのセンサー状態が格納されます。このアレイは、ヘッダ ファイル CSA_EMC.h で定義されています。

センサー状態は、ユーザー モジュール API CSA_bIsAnySensorActive()により更新されます。

3.4 CSA_EMC ユーザー モジュールのパラメーター

CSA_EMC ユーザー モジュールのパラメーターは、高レベル パラメーターまたは低レベル パラメーターとして分類されます。CSA_EMC ユーザー モジュールのパラメーターのリストおよびその分類方法については、図 3-4 を参照してください。

図 3-4. PSoC Designer - CSA_EMC パラメーター ウィンドウ



3.4.1 高レベル パラメーター

3.4.1.1 指の閾値

このパラメーターは、センサーのアクティブ／非アクティブ状態を判断するユーザー モジュールによって使用されます。センサーの差分カウント値が指の閾値を超えると、センサーはアクティブと判断されます。この定義は、ヒステリシスがゼロに設定され、デバウンスが 1 に設定されていることが前提となります。

可能な値は 3～255 です。

推奨値は、「CSA_EMC ユーザー モジュール – チューニング ガイド」を参照してください。

3.4.1.2 ヒステリシス

ヒステリシス設定により、システム ノイズによってオン状態のセンサーにチャタリングが起きることを防ぎます。ヒステリシスの関数は式 4 で計算されます。この式では、デバウンスが 1 に設定されていることが前提となります。

図 3-5. ヒステリシスがゼロに設定されている場合のセンサー状態対差分カウント

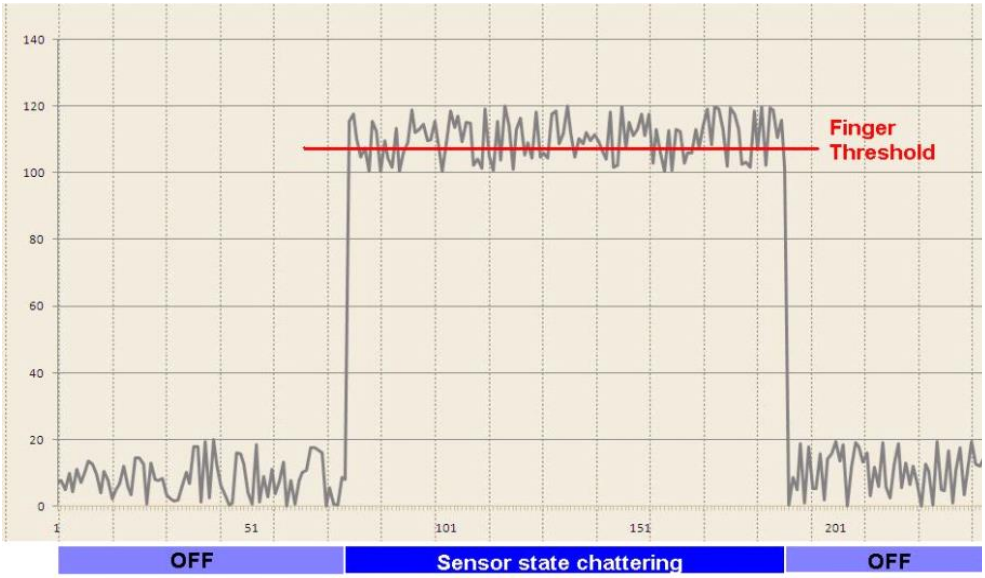
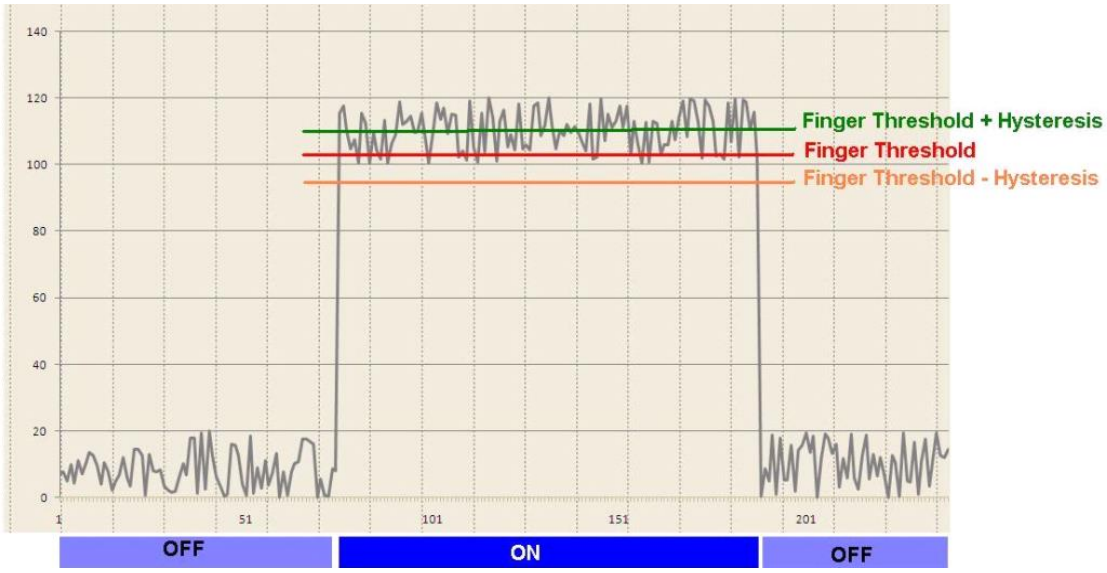


図 3-6. ヒステリシスが設定されている場合のセンサー状態対差分カウント



差分カウント ≥ 指閾値 + ヒステリシスの場合、　センサー状態 = ON

差分カウント ≥ 指閾値 - ヒステリシスの場合、　センサー状態 = OFF　　式 4

可能な値は 0～255 です。

推奨値は、「CSA_EMC ユーザー モジュール – チューニング ガイド」を参照してください。

3.4.1.3 デバウンス

デバウンスによって、raw カウントのスパイクでセンサー状態がオフからオンに変わらないようになります。センサー状態をオフからオンへ遷移するには、指定されたサンプル数で、差分カウント値が指の閾値 + ヒステリシスのレベルを上回るようにする必要があります。

可能な値は 1～255 です。1 にセットするとデバウンスは起こりません。

推奨値は、「CSA_EMC ユーザー モジュール – チューニング ガイド」を参照してください。

3.4.1.4 ベースライン更新閾値

前に説明したように、ベースライン変数には raw カウント値の段階的な変動記録が保持されます。すなわち、ベースライン変数は、入力 raw カウント値を受け取るデジタル ローパス フィルターの出力を保持します。ベースライン更新閾値パラメーターはこのローパス フィルターの時定数を調整するために使用されます。

ベースライン更新閾値は、このフィルタの時定数に正比例します。ベースライン更新閾値が高くなるほど、時定数が高くなります。

可能な値は 1～255 です。

推奨値は、「CSA_EMC ユーザー モジュール – チューニング ガイド」を参照してください。

3.4.1.5 ノイズ閾値

ノイズ閾値は、ユーザー モジュールが raw カウントのノイズ カウントの上限を知る上で役立ちます。個別センサーの場合、ベースライン更新アルゴリズムは、raw カウントがベースライン値を上回りこれらの差分がこの閾値を上回ると停止します。

スライダー センサーの場合、差分カウントがノイズ閾値を超えるとセントロイド計算は停止されます。

可能な値は 3～255 です。ユーザー モジュールが正常に動作するために、ノイズ閾値を [指閾値 － ヒステリシス] より高く設定しないでください。

推奨値は、「CSA_EMC ユーザー モジュール – チューニング ガイド」を参照してください。

3.4.1.6 負のノイズ閾値

負のノイズ閾値は、ユーザー モジュールが raw カウントのノイズ カウントの下限を知る際に役立ちます。ベースライン更新アルゴリズムは、raw カウントがベースライン値を下回りこれらの差分がこの閾値を上回ると停止します。

可能な値は 0～255 です。

推奨値は、「CSA_EMC ユーザー モジュール – チューニング ガイド」を参照してください。

3.4.1.7 低ベースライン リセット

低ベースライン リセット パラメーターは、負のノイズ閾値パラメーターと併用されます。指定されたサンプル数で、サンプル カウント値がベースライン値か負のノイズ閾値を引いた値を下回ると、ベースライン値は新しい raw カウント値に設定されます。これは基本的に、ベースライン値をリセットする必要がある異常に低いサンプルをカウントします。起動時に指が置かれている状態の補正をするために使用されます。

可能な値は 0～255 です。

推奨値は、「CSA_EMC ユーザー モジュール – チューニング ガイド」を参照してください。

3.4.1.8 センサーの自動リセット

センサーの自動リセットパラメーターによって、ベースライン値は常に更新されるのか、それとも差分カウントがノイズ閾値を下回った場合にのみ更新されるかが決定されます。

センサーの自動リセットがイネーブルにされていると、ベースライン値は常に更新されます。この設定により、センサーの最大時間は制限されますが、センサーに何も触れないのに raw カウントが誤って上昇した場合に、センサーが恒久的にオンになることを妨ぐことができます。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。

センサーの自動リセットが無効化されていると、ベースライン値は差分カウントがノイズ閾値パラメーターを下回った場合にのみ更新されます。

可能な値は、Enabled と Disabled です。

推奨値は、「CSA_EMC ユーザー モジュール – チューニング ガイド」を参照してください。

3.4.2 低レベル パラメーター

CSA_EMC ユーザー モジュールには、高レベル パラメーターに加えて低レベル パラメーターがあります。これらは CSA_EMC センシング方式固有のパラメーターで、センサーから raw カウント データを取得する方法を決定します。

3.4.2.1 整定時間

整定時間パラメーターはソフトウェアの遅延を制御し、C_{INT} コンデンサの電圧を積分平衡電位にすることが可能です。

可能な値は 2～255 です。

推奨値は、「CSA_EMC ユーザー モジュール – チューニング ガイド」を参照してください。

3.4.2.2 耐性レベル

耐性レベル パラメーターは、ノイズ耐性レベルを定義します。使用できるオプションは低と高です。

高オプションを選択すると、高ノイズ環境でのパフォーマンスが改善されます。ただし、メモリ使用量が増えるため、サポートされるセンサーの最大数は減ります。

また、高に設定の時に要するスキャン時間は低に設定した場合の 3 倍になります。

3.4.2.3 スペクトラム拡散

スペクトラム拡散を有効にすると、外部 RF 妨害がある場合の CSA_EMC アルゴリズムのパフォーマンスが改善されます。アルゴリズムの放射特性も改善されます。

可能な値は、Enabled と Disabled です。

3.4.2.4 raw データ メディアン フィルター

このメディアン フィルターは、センサーから最新のサンプル 3 つを調べ、中央値を報告します。これは、短いノイズ スパイクを除去するために使用されます。1 つのサンプルの遅延が発生するため、デフォルトでは無効化になっています。

可能な値は、Enabled と Disabled です。

3.4.2.5 raw データ IIR フィルター

この無限インパルス応答 (IIR) フィルターは、変換の結果 (raw カウント) 内のノイズを減少させます。これは、デフォルトでは無効になっています。

可能な値は、Enabled と Disabled です。

3.4.2.6 raw データ IIR フィルター係数

この設定は、raw カウント IIR フィルターの係数です。

可能な値は、2 (½ 前回のサンプル + ½ 現在のサンプル) と 4 (¾ 前回のサンプル + ¼ 現在のサンプル) です。

3.4.2.7 クロック

クロック パラメーターは、スイッチ Sw1 と Sw2 を制御する非重複プリチャージ クロックの周波数を設定します。Sw1 と Sw2 は、センサー コンデンサの接続を AMUX バスとグランド間で交互に切り替える場合に使用します。最大信号にするには、各サイクル中にセンサー コンデンサが完全に充電および放電されるように、プリチャージ クロックの周波数を設定する必要があります。

可能な値は、IMO、IMO/2、IMO/4、IMO/8 です。

4. ユーザー モジュールによる CapSense 性能のチューニング

最適なユーザー モジュール パラメーターの設定は、基板レイアウト、ボタンのサイズ、オーバーレイ素材、アプリケーション要件によって異なります。これらの要因については、「設計の注意事項」で説明します。チューニングとは、安定で信頼できるセンサー操作のための最適なパラメーター設定を特定するプロセスです。

4.1 一般的な注意事項

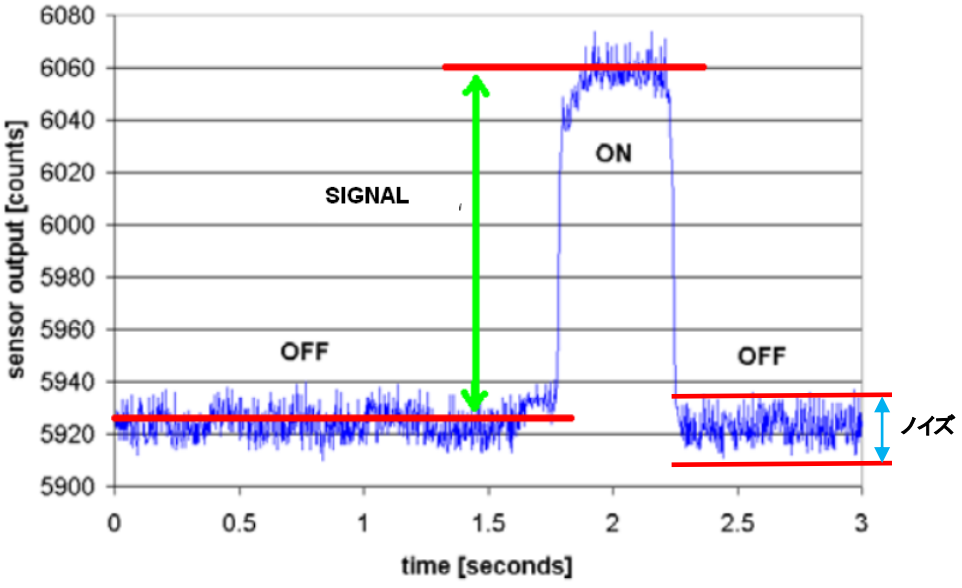
4.1.1 信号、ノイズ、および SNR

適切にチューニングされた CapSense システムによって、オンとオフのセンサー状態が確実に識別されます。このレベルの性能を得るには、CapSense 信号が CapSense ノイズよりかなり大きくなる必要があります。CapSense 信号は、信号対ノイズ比 (SNR) という量を使用して CapSense ノイズと比較されます。CapSense に対する SNR の意味を説明する前に、まず、タッチ検知において信号とノイズとは何かを定義する必要があります。

4.1.1.1 CapSense 信号

CapSense 信号とは、図 4-1 に示すように、センサー上に指を置いたときのセンサー反応の変化です。センサーの出力は、センサー静電容量を追跡する値のデジタル カウンターです。この例では、センサーに指を置かない場合の平均レベルは 5925 カウントです。センサーに指を置いた場合の平均出力は 6060 カウントに増えます。CapSense 信号は、指によるカウントの変化を追跡するため、信号 = 6060 – 5925 = 135 カウントとなります。

図 4-1. CapSense の信号とノイズ



4.1.1.2 CapSense ノイズ

CapSense ノイズは、図 4-1 に示されているように、指の接触がない時のセンサー反応におけるピーク ツー ピークの変化です。この例では、指を置かない場合の出力波形は、最小で 5912 カウント、最大で 5938 カウントです。ノイズとはこの波形の最小値と最大値の差分であるため、ノイズ = 5938 – 5912 = 26 カウントとなります。

4.1.1.3 CapSense SNR

CapSense SNR とは信号とノイズの単純比です。この例では、信号が 135 カウントであり、ノイズが 26 カウントである場合、SNR は 135:26 で、5.2:1 に約分されます。CapSense の推奨最小 SNR は 5:1 で、すなわち信号値がノイズ値より 5 倍大きいです。一般的に、フィルターはノイズを減少させるためにファームウェアに実装されます。詳細は「ソフトウェアのフィルター処理」を参照してください。

4.1.2 ベースライン更新閾値の検証の重要性

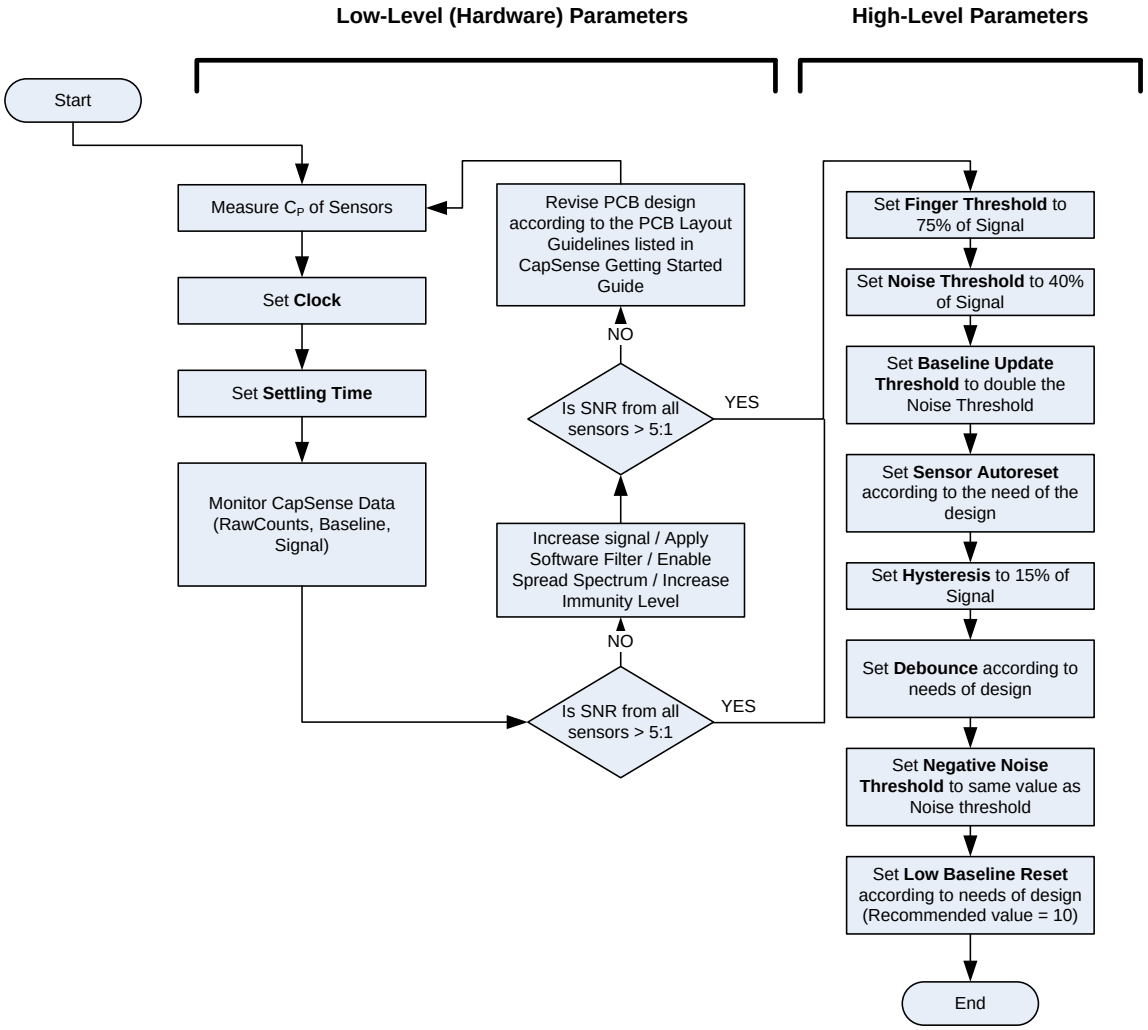
温度と湿度はどちらも、平均カウント数の経時的変動の原因となります。ベースライン値は CapSense 測定の参照カウント レベルであり、環境からの影響補正に重要な役割を果たします。指あり状態と指なし状態といった高レベル決定は、ベースライン値で設定される参照レベルに基づきます。各センサーそれぞれ別の寄生容量があるため、各静電容量センサーには独自のベースラインがあります。

ベースラインは、ベースライン更新閾値パラメーターで設定された速度でカウントの変化を追跡します。更新速度が目的に合致していることを確認します。更新速度が速すぎると、ベースライン値は指による全ての変化を補正するため、動いている指は検出されません。更新速度が遅すぎると、比較的ゆっくりの環境変化が指と間違えられる可能性があります。開発時には、ベースライン更新閾値パラメーター設定の検証をお奨めします。

4.2 CSA_EMC ユーザー モジュール – チューニング ガイド

図 4-2 は、CSA_EMC パラメーターのチューニング プロセスを示したフローチャートです。CSA_EMC ユーザー モジュールのパラメーターは、低レベル (ハードウェア) パラメーターと高レベル パラメーターの 2 つカテゴリに大きく分類できます。これらのカテゴリのパラメーターは、さまざまな方法で静電容量センシング システムの動作に影響します。ただし、センサー感度はハードウェア パラメーター設定と多くの高レベル パラメーター設定で決まるため、各センサーの感度の間には補完関係があります。ハードウェア パラメーターを変更する場合は、この点を考慮して、対応する高レベル パラメーターがそれに応じて調整されることを確認する必要があります。CSA_EMC ユーザー モジュールのパラメーターのチューニングは、常にハードウェア パラメーターから開始する必要があります。

図 4-2. CSA_EMC ユーザー モジュール パラメーターのチューニング フローチャート



4.3 推奨 CINT 値

積分コンデンサ CINT の推奨値 1.2nF でチューニング プロセスを開始します。チューニング プロセス中に、センサー信号が 5:1 SNR を得るために不十分であると判断する場合は、CINT を増加できます。CINT の推奨最大値は 5.6nF です。温度の変化時に CINT が安定するために X7R または NPO タイプのコンデンサの使用をお奨めします。

4.4 センサーCP の測定

チューニング プロセスの最初のステップは、センサーの寄生容量 (CP) の測定です。段階を追った手順は、以下のとおりです。

1. **CPU_CLK** を SysClk/2 に設定する
2. **クロック**を IMO/8 に設定する
3. **整定時間**を 255 に設定する
4. 特定のセンサーに対してアルゴリズムで設定された IDAC コードをリードバック。値はアレイ CSA_EMC_baDACCodeBaseline[]に格納される
5. IDAC コードに対応する IDAC 電流を測定する

次のコードで PSoC Designer プロジェクトを作成します。このコードでは、IDAC がポート ピン P1[4]に接続されます。

```
//configure P1[4] to HI-Z
PRT1DM0 &= ~ (1<<4);
PRT1DM1 |= (1<<4);
//connect P1[4] to analog mux bus
MUX_CR1 = (1<<4);
// set IDAC to read back IDAC Code
IDAC_D = <IDAC CODE>
// turn ON IDAC
CS_CR2 = 0xD0;
```

ピン P1[4] とグランドとの間に電流メーターを配置し、電流を測定します。$I_{MEASURED}$ をその値に設定します。

6. 式 5 を使用して、C_P を計算する

$$C_P = \frac{I_{MEASURED}}{(\frac{IMO}{8} \times 1.3)} \qquad \text{式 5}$$

または、センサーC_P は LCR メーターで測定することもできます。LCR メーターの 1 つの端子をセンサー ピンに、もう 1 つを GND に接続し、C_P を測定します。

4.5 CSA_EMC クロックの予測

プリチャージ クロックは、CSA_EMC クロックのユーザー モジュールのパラメーターで設定されます。これは、CSA_EMC 設計を正しくチューニングする上で、最も重要なハードウェア UM パラメーターです。CSA_EMC クロックの推奨値を、表 4-1 に示します。

表 4-1. C_P と IMO に基づいた CSA_EMC クロック設定

	CSA_EMC クロック	
C_P (pF)	IMO = 12MHz	IMO = 6MHz
< 5	IMO	IMO
5 ～ 10	IMO/2	IMO
10～ 15	IMO/4	IMO/2
15 ～ 20	IMO/4	IMO/2
20～ 25	IMO/8	IMO/4
25 ～ 30	IMO/8	IMO/4
30～ 35	IMO/8	IMO/4
35 ～ 40	IMO/8	IMO/4
40～ 45	IMO/8	IMO/4
45 ～ 50	IMO/8	IMO/8

4.6 整定時間の設定

整定時間パラメーターの最小値は、式 6 で予測します。

$$Settling\ Time = \frac{5 \times C_{INT}}{(Clock \times C_P \times 25 \times \frac{1}{FCPU})} \qquad \text{式 6}$$

ここで、

C_{INT} = 積分コンデンサの値

クロック = プリチャージ クロックの周波数 (CSA_EMC クロック)

C_P = センサーの寄生容量値

F_{CPU} = CPU クロック周波数

4.7 CapSense データの監視

「CapSense データ表示ツール」を参照してください。

4.8 SNR の向上方法

SNR を向上するには、ノイズ カウントを減らすか、信号を増加します。

4.8.1 ノイズの削減

SNR を向上する方法の 1 つは、ノイズ カウントを減らすことです。以下のオプションのいずれかを使用できます。

- ソフトウェア フィルターを使用 - 詳細は、「ソフトウェアのフィルター処理」を参照してください。
- スペクトラム拡散の有効化—詳細は、「スペクトラム拡散」を参照してください。
- 耐性レベルを増加 - 詳細は、「耐性レベル」を参照してください。

4.8.2 信号の増大

SNR は信号を強化して改善することもできます。以下の 2 つの方法のどちらかを使用します:

- マクロ CSA_EMC_BASELINE で定義した値を高くします。このマクロは *CSA_EMC.inc.* ファイルにあります。初期設定では、マクロには値 0x0800 が割り当てられます。
- C_{INT} コンデンサの値を増加します。

4.9 チューニングの例

このセクションでは、以下のチューニング ガイドラインに従って、単一の CapSense センサーをチューニングする方法の例を示します。この例では、「PSoC Designer」と「CapSense データ表示ツール」に習熟していることを前提としています。詳細については、「PSoC Designer ユーザー ガイド」および「AN2397 - CapSense データ表示ツール」を参照してください。

4.9.1 ハードウェア要件

チューニングはご使用のボードで実行するか、標準の Cypress CapSense UCC 基板 (リンクは下記) を使用して実行します。

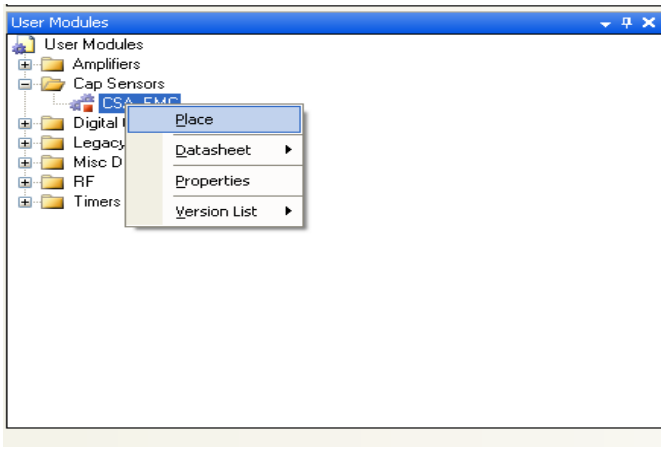
- CY3280-20x34 ユニバーサル CapSense コントローラー
- CY3280-BSM シンプル ボタン モジュール キット

以下に段階的に示すチューニング手順は、ポート ピン P1[4]で CapSense センサーを、ポート ピン P0[3]で積分コンデンサを使用します。

4.9.2 段階的なチューニングガイド

1. ターゲット デバイスとして、CY8C20x34 で新しい PSoC Designer プロジェクトを作成します。
2. **User Module Catalog** ウィンドウに移動します。CSA_EMC ユーザー モジュールを Cap Sensors カテゴリから選択・配置します。

図 4-3. CSA_EMC ユーザー モジュールの選択・配置



3. ワークスペース エクスプローラで CSA_EMC ユーザー モジュールを右クリックし、CapSense ウィザードにアクセスします。センサーと C_{INT} にピンを割り当てます。

図 4-4. CSA_EMC ユーザー モジュールを右クリックし、CapSense ウィザードを起動

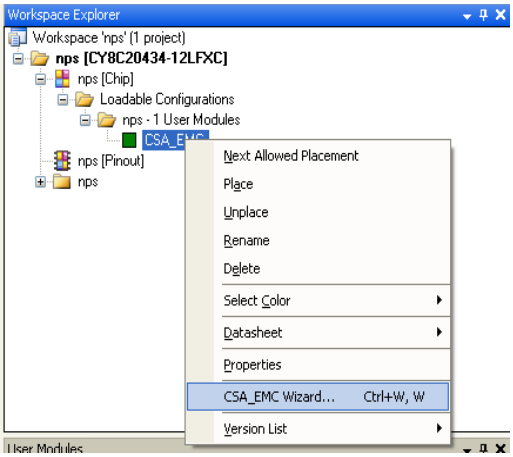
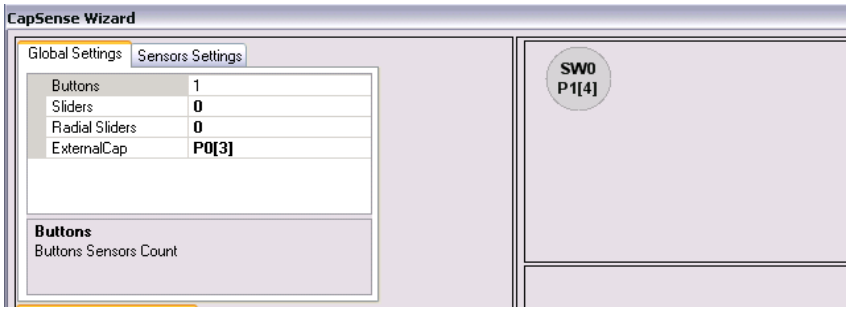
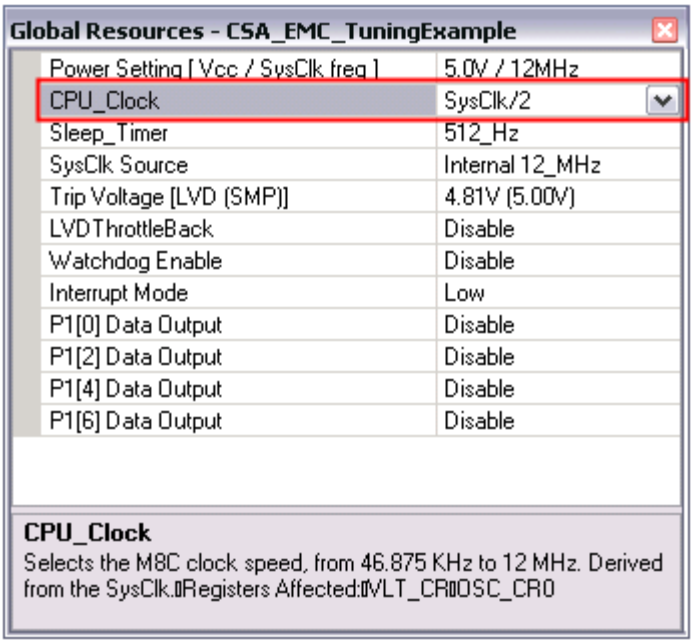


図 4-5. CapSense ウィザード(センサー ピン = P1[4]、C_{INT} ピン = P0[3])



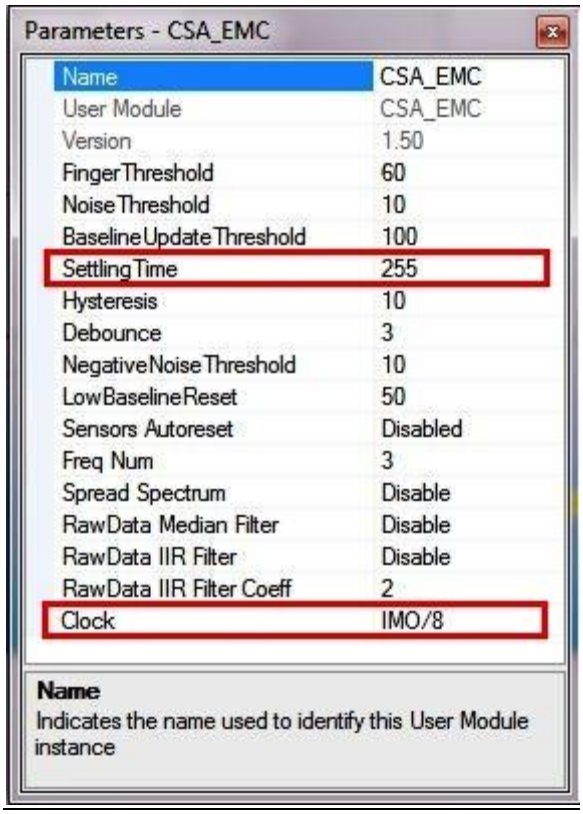
4. **Global Resourses** ウィンドウで、**CPU_CLK** を SysClk/2 に等しくなるように設定します。設計ニーズに合わせて、**Global Resourses** ウィンドウの残りのパラメーターを設定します。

図 4-6. CPU_CLK = SysClk/2 に設定



5. **Parameters - CSA_EMC** ウィンドウで **Clock** = IMO/8 および **Settling Time** = 255 と設定します。その他すべてのパラメーターのデフォルト値を維持します。

図 4-7. クロック = IMO/8、整定時間 = 255 に設定



6. デジタル通信インターフェース (I²C または SPI) を利用し、CapSense 表示ツール (MultiChart またはブリッジ制御パネル) を使用して CSA_EMC_baDACCodeBaseline[] の IDAC コードをリードバックします。詳細については、「AN2397 - CapSense データ表示ツール」を参照してください。

   サンプル コード:

```
void main(void)
{
    M8C_EnableGInt;
    CSA_EMC_Start();
    …
        …
    while(1)
    {
    // OUTPUT CSA_EMC_baDACCodeBaseline[0] <- IDAC code for sensor 0;
    }
}
```

このステップで見つかったリードバック コードを 0x08 にします。

7. 0x08 の I_{DAC} コードに対応する I_{DAC} 電流を測定します。

   サンプル コード:

```
void main(void)
{
        //configure P1[4] to HI-Z
        PRT1DM0 &= ~ (1<<4);
        PRT1DM1 |= (1<<4);
        //connect P1[4] to analog mux bus
        MUX_CR1 = (1<<4);
        // set IDAC to read back IDAC Code
        IDAC_D = 0x08;
        // turn ON IDAC
        CS_CR2 = 0xD0;
}
```

電流の測定方法の詳細については、センサーCP の測定のステップ 5 を参照してください。

測定した電流を 23µA とします。

8. C_P を計算します。

   式 5 から、$C_P = I_{MEASURED} / ((IMO/8) * 1.3)$

   以下の値を代入する場合:

   $I_{MEASURED}$ = 23µA

   IMO = 12MHz (この例では IMO を 12MHz に設定)

   結果は C_P = 12 pF です。

9. CSA_EMC クロックを計算します。

   表 4-1 から、C_P = 12 pF および IMO = 12MHz の場合、推奨するクロックは IMO/4 です。

10. 整定時間を計算します。

    式 5 から:

$$Settling\ Time = \frac{5 \times C_{INT}}{(Clock \times C_P \times 25 \times \frac{1}{FCPU})}$$

    以下の値を代入する場合:

    C_{INT} = 1.2nF

クロック = IMO/4 = 3MHz

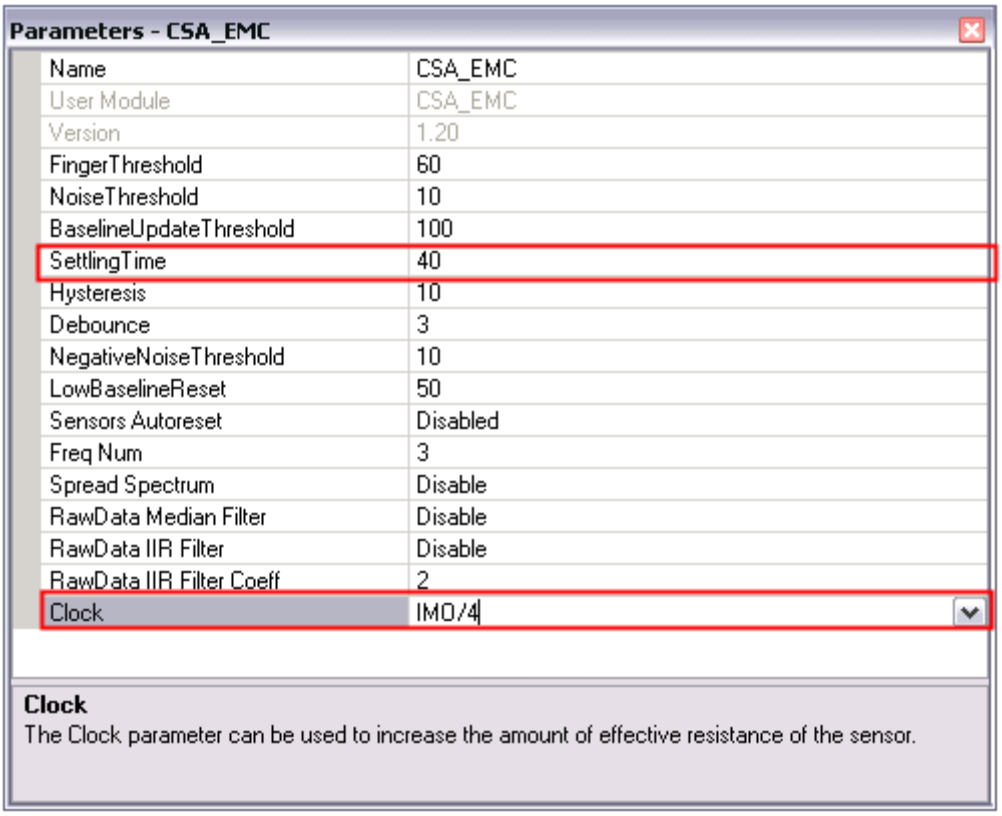
C_P = 12pF

F_{CPU} = SysClk/2 = 12MHz/2 = 6MHz (この例では、SysClk は 12MHz に設定します)

結果は整定時間 = 40 です。

11. **Parameters - CSA_EMC** ウィンドウでクロックおよび整定時間の値を更新します。

図 4-8. クロック = IMO/4、整定時間 = 40 と設定



12. 新しい設定を行って、SNR を測定します。

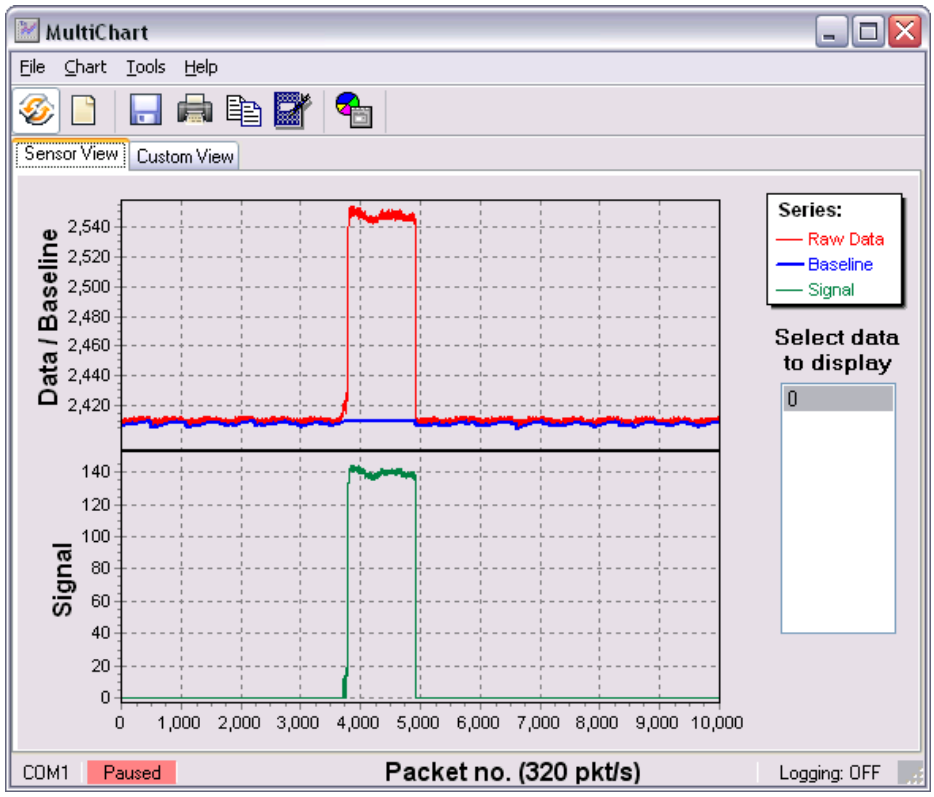
図 4–9 は、観測された raw カウント、ベースライン、および信号レベルを示します。

信号 = 140 カウント (概数)

ノイズ = 10 カウント (概数)

このため、SNR は 14 です。これは 5 よりも多くなっています。

図 4-9. 信号 = 140 カウント、ノイズ = 10 カウント



13. 高レベル パラメーターを設定してください。

    指閾値 = 信号の 75% = 0.75 × 140 = 105

    ノイズ閾値 = 信号の 40% = 0.40 × 140 = 56

    ベースライン更新閾値 = 2 × ノイズ閾値 = 112

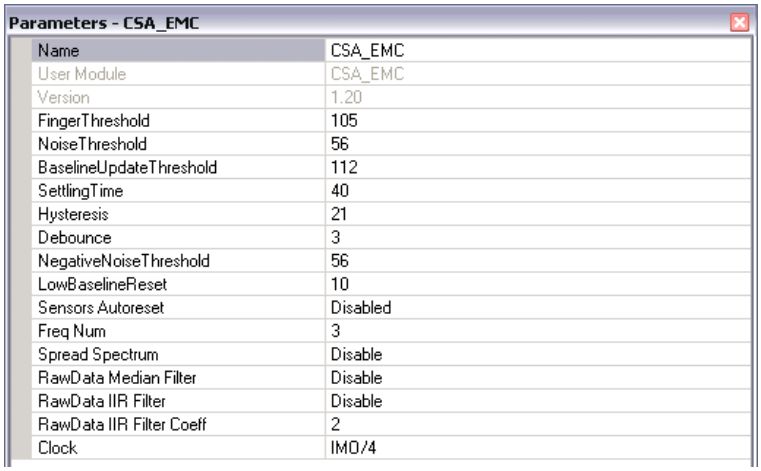
    ヒステリシス = 信号の 15% = 0.15 × 140 = 21

    負のノイズ閾値 = ノイズ閾値 = 56

    低ベースライン リセット = 10

14. 以上の計算に基づいて、CSA_EMC パラメーター値を更新します。

図 4-10. 更新された CSA_EMC パラメーター ウィンドウ



15. 新規設定を行って、*main.c* ファイルに必要な設計別のコードを作成してプロジェクトを再生成します。

5. 設計の注意事項

静電容量式タッチ センサー技術をアプリケーションで設計する際に、CapSense デバイスがより大きなフレームワーク内に存在していることを 覚えておくことが必須となります。プリント基板レイアウトからユーザーインターフェース、そしてエンドユースのオペレーティング環境にいたる全レベルの細部に注意を払えば、しっかりとした信頼のおけるシステム性能がもたらされます。詳細については、「CapSense 入門」を参照してください。

5.1 オーバーレイの選択

「CapSense の原理」モデルで、CapSense に相当するモデルを、指の静電容量の式と一緒に示しました:

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

ここで、

ε_0 = 真空の誘電率

ε_r = オーバーレイの比誘電率

A = 指とセンサー パッドの重複面積 (mm^2)

D = オーバーレイの厚さ (mm)

CapSense の信号強度を高めるには、より誘電率が高いオーバーレイ素材を使用し、オーバーレイの厚さを薄くし、ボタン直径を大きくします。

表 5-1. オーバーレイ素材の絶縁耐力

素材	絶縁破壊電圧 (V/mm)	12 kV でのオーバーレイの最小の厚さ (mm)
空気	1200～2800	10
木材 (乾燥)	3900	3
ガラス (一般的)	7900	1.5
ガラス – ホウケイ酸塩 (Pyrex®)	13,000	0.9
PMMA プラスチック (プレキシグラス®)	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 (高圧メラミン化粧板)	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

導電性の素材は、電場パターンを妨げるため、オーバーレイとしては使用できません。このため、金属粒子を含有する塗料をオーバーレイで使用しないでください。

オーバーレイを CapSense プリント基板に接合するために、通常は接着剤を使用します。3M™ の 200MP と呼ばれる透明なアクリル系接着剤は CapSense アプリケーションでの使用に適しています。この特殊な接着剤は裏に紙が付いたテープロール形状で販売されています (3M™商品番号 467MP および 468MP)。

5.2 ESD 保護

安定した ESD 耐性は、慎重なシステム設計から生まれた当然の結果です。最終製品、特にユーザーインターフェースにどのような接触放電が起こるかを熟慮することで、CapSense コントローラーに損傷を与えることなく 18-kV の放電現象に耐えることが可能です。

CapSense コントローラー ピンは直接的な 2-kV 放電現象に耐えることが可能です。ほとんどの場合、オーバーレイの素材がコントローラー ピンへの十分な ESD 保護を提供します。表 5-1 は、IEC 61000-4-2 で指定されているように、CapSense センサーを 12kV 放電から保護するために必要となる、さまざまなオーバーレイの厚さを一覧表示しています。オーバーレイの素材が十分な ESD 保護を提供しない場合、対応策は次の順番で適用されます: 防止、リダイレクト、クランプ。

5.2.1 防止

接触面のすべてのパスが、潜在的な高電圧接触よりも高い絶縁破壊電圧を備えていることを確認してください。またシステム内の、CapSense コントローラーと ESD 発生源となり得る部分間の距離がいつでも適切であるように設計します。適切な距離を維持できない場合は、絶縁破壊電圧の高い素材による保護レイヤを ESD ソースと CapSense コントローラーの間に設けてください。5mil の厚さの Kapton®テープによる層は 18kV に耐久します。

5.2.2 リダイレクト

基板のコンポーネントの密度が高い場合は、放電現象を避けることは難しいかもしれません。この場合、放電の発生場所を制御することで CapSense コントローラーを保護できます。標準的な手順としては、シャーシの接地に接続された回路ボードの外周部に保護リングを配置します。「プリント基板レイアウトのガイドライン」でお勧めするように、ボタンやスライダー センサーの周囲にハッチング グランド面を施すことによってセンサーと CapSense コントローラーへの ESD の影響を回避することができます。

5.2.3 クランプ

CapSense センサーは意図的にタッチ面に近接して配置されているため、放電パスのリダイレクトは実践的でないかもしれません。この場合には、直列抵抗器や特殊用途の ESD 保護デバイスを使用するのが適切です。

推奨する直列抵抗は 560Ω です。

より効果的な方法は、専用の ESD 保護デバイスを脆弱な配線上に置くことです。CapSense 用の ESD 保護デバイスは低静電容量です。表 5-2 に、CapSense コントローラー向けの推奨デバイスを示します。

表 5-2. CapSense 用の推奨低静電容量 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	接触放電の最高限度	空中放電の最高限度
メーカー	型番				
Littlefuse	SP723	5pF	2nA	8kV	15kV
Vishay	VBUS05L1-DD1	0.3pF	0.1µA<	±15kV	±16kV
NXP	NUP1301	0.75pF	30nA	8kV	15kV

5.3 電磁環境適合性 (EMC) の注意点

5.3.1 放射性干渉

放射性電気エネルギーはシステム測定に影響を与え、さらにプロセッサ・コアのオペレーションに影響を与える可能性もあります。干渉は、CapSense センサー トレースを通して、さらにその他のデジタルまたはアナログ入力を通してプリント基板レベルで PSoC チップに入ります。以下に RF 干渉の影響を少なくするためのレイアウトガイドラインを示します:

- **グランド面**: プリント基板にグランド面を設けます。
- **直列抵抗**: CapSense コントローラー ピンから 10mm 以内に直列抵抗を配置します。
  - CapSense 入力ラインで推奨する抵抗値は 560Ω
  - I^2C や SPI などの通信回線の推奨直列抵抗は 330Ω
- **配線の長さ**: 可能な限り配線を短くします。
- **電流ループのエリア**: 電流の帰路を短くします。固体を充填する代わりに、ハッチング グランドをセンサーとトレースの 1 cm 以内に取り付け、寄生容量の影響を低減させます。
- **RF 源の位置**: LCD インバータおよびスイッチング電源 (SMPS) のようなノイズ源を CapSense 入力から隔てるために隔離手段をとります。電源のシールドは干渉を防ぐもう一つの良くある技術です。

5.3.2 放射エミッション

CapSense センサーからの放射エミッションを低減させるには、スイッチト コンデンサのクロックに低周波数を選択してください。このクロックは、プリスケーラ オプションを使用するファームウェアで制御しています。プリスケーラ値を高くすると、スイッチング クロックの周波数が低くなります。

5.3.3 伝導イミュニティおよびエミッション

他のシステムとの電線による接続を通じてシステムに入ったノイズは伝導ノイズと呼ばれます。これらの接続には電源や通信ラインを含みます。CapSense コントローラーは低消費電力デバイスなため、伝導エミッションを回避しなくてはいけません。以下のガイドラインに従って伝導エミッションおよびイミュニティを低減することができます。

- データシートで推奨しているようにデカップリング コンデンサを使用します。
- システム電源への入力に双方向性のフィルターを追加します。導電性エミッションおよびイミュニティの両方に効果的です。π フィルターは、電源ノイズが感度の高い部品に影響を与えるのを防ぐことができます。その一方で、部品それ自体のスイッチング ノイズがカップリングから電力面に逆流することも防止します。
- CapSense コントローラーのプリント基板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することが必要です。
- 電源供給や通信回線の周囲にフェライトビーズを配置し、高周波ノイズを除去することができます。

EMC の考慮に関する詳細は、「CapSense 入門」を参照してください。

5.4 ソフトウェアのフィルター処理

ソフトウェア フィルターは、高レベルのシステム ノイズに対処する 1 つの方法です。表 5-3 に、CapSense に有効であると認められたフィルターの種類を一覧表示します。

表 5-3. CapSense フィルターの表

タイプ	説明	応用
アベレージ	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)	電源からの周期的なノイズ
IIR	RC フィルターに類似したステップ レスポンスを備えた、有限インパルス応答フィルター (フィードバック付き)	高周波ホワイト ノイズ(1/f ノイズ)
Median (メジアン)	サイズ N のバッファからメジアン入力値を計算する非線形フィルター	モーターおよび電源切り替えに伴って出るスパイク ノイズ
Jitter (ジッタ)	前の入力に基づいて、電流の入力を制限する非線形フィルター	厚いオーバーレイからのノイズ (SNR < 5:1) ; 特にスライダーのセントロイド データに役立つ
イベント ベース	センサー データで観察されたパターンへの事前定義された応答を発生させる非線形フィルター	CapSense データ送信をブロックする非接触イベント中によく使用される
ルール ベース	センサー データで観察されたパターンへの事前定義された応答を発生させる非線形フィルター	タッチ表面の通常操作中によく使用され、偶発的なマルチボタン選択などの特殊なシナリオに対応

表 5-4 はさまざまなソフトウェア フィルターで必要な RAM およびフラッシュの詳細を示しています。各フィルター種類に必要なフラッシュの大きさは、コンパイラの性能に依存しています。ここで一覧表示している要件は、ImageCraft コンパイラと ImageCraft Pro コンパイラ双方に該当します。

表 5-4. RAM およびフラッシュの要件

フィルターの種類	フィルターの順序	RAM (センサー毎のバイト)	フラッシュ (バイト) ImageCraft コンパイラ	フラッシュ (バイト) ImageCraft Pro コンパイラ
アベレージ	2-8	6	675	665
IIR	1	2	429	412
	2	6	767	622
Median (メジアン)	3	6	516	450
	5	10	516	450
raw カウントのジッタ フィルター	N/A	2	277	250
スライダー セントロイドのジッタ フィルター	N/A	2	131	109

ソフトウェア フィルターの詳細については、「CapSense 入門」を参照してください。

5.5 消費電力

5.5.1 システム デザインの推奨事項

多くのデザインで、消費電力を最小限に抑えることは重要な目標です。CapSense 静電容量タッチ センサー システムの消費電力を低減するには、いくつかの方法があります。

- GPIO 駆動モードを低電力用に設定する
- 高電源ブロックを切る
- 低電力用に CPU の速度を最適化する
- 低い V_{DD} で動作させる

これらの方法に加えて、スリープ スキャン方法を適用することが大変有効です。

5.5.2 スリープ・スキャン方式

5.5.2.1 平均電力

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

長期的にデバイスが消費する平均電流は、以下の式を使用して計算することができます。

$$I_{AVE} = \frac{(I_{Act} \times t_{Act}) + (I_{Slp} \times t_{Slp})}{T} \qquad \text{式 7}$$

ここで、

I_{AVE} = デバイスの平均電流

I_{Act} = デバイスの有効電流

t_{Act} = デバイスのアクティブ時間

I_{Slp} = デバイスのスリープ電流

t_{Slp} = デバイスのスリープ時間

$T = t_{Act} + t_{Slp}$

デバイスが消費した平均電力は、次のように計算することができます。

$$P_{AVE} = V_{DD} \times I_{AVE} \qquad \text{式 8}$$

ここで、

P_{AVE} = デバイスの平均電力

V_{DD} = デバイスの電源電圧

I_{AVE} = デバイスの平均電流

5.5.2.2 応答時間対消費電力

式 8 で図示しているように、平均消費電力は I_{AVE} または V_{DD} を低下させて削減することができます。I_{AVE} はスリープ時間を増やして、削減することができます。スリープ時間をきわめて高い値にすると、CapSense ボタン応答時間に影響が及びます。結果として、スリープ時間はシステム要件に基づいている必要があります。

アプリケーションで、消費電力と応答時間は考慮すべき重要なパラメーターである場合、連続スキャンとスリープ スキャン モードの双方を組み合わせて最適化する方法を使用できます。この方法では、デバイスはその時間の大半をスリープ スキャン モードで使い、前のセクションで説明したように、定期的にセンサーをスキャンしてスリープ状態に入り、消費電力が少なくなります。ユーザーがセンサーに触れてシステムを操作すると、デバイスが連続スキャン モードに入って、スリープ状態に入

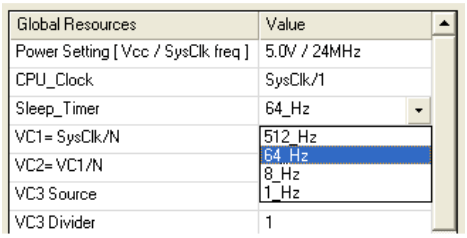
らずにセンサーを連続的にスキャンし、応答時間が大幅に向上します。デバイスは、指定されたタイムアウト期間の間、連続スキャン モードのままとなります。ユーザーがこのタイムアウト期間内にセンサーを操作しない場合、デバイスはスリープ スキャン モードに戻ります。

5.5.2.3 平均消費電力の測定

以下に、スリープ スキャン方法を使用するときの平均消費電力を判定する方法について説明します。

1. スリープ状態に入らずに全センサーをスキャンするプロジェクトを作成します(連続スキャン モード)。センサーをスキャンする前に、コードにピン トグル機能を組み込みます。出力ピンの状態を切り替えることで、スコープで追跡可能なタイム マーカとして機能させることができます。
2. プロジェクトを CapSense デバイスにダウンロードし、消費電流を測定します。測定した電流を IAct に割り当てます。
3. データシートからスリープ電流情報を入手し、それを ISlp に割り当てます。
4. 出力ピンの切り替えをオシロスコープで監視し、2 つの切り替えの間の時間間隔を測定します。これにより有効時間が分かります。この値を tAct に割り当てます。
5. スリープ スキャンをプロジェクトに適用します。スリープ スキャン サイクルの時間間隔 T を設定するには、図 5-1 に示すように、Global Resources ウィンドウでスリープ タイマーの頻度を選択します。
6. スリープ スキャン サイクル周期から有効時間を減算して、スリープ時間を求めます。tSlp = T – tAct。
7. 式 7 を使用して平均電流を計算します。
8. 式 8 を使用して平均消費電力を計算します。

図 5-1. Global Resources ウィンドウ

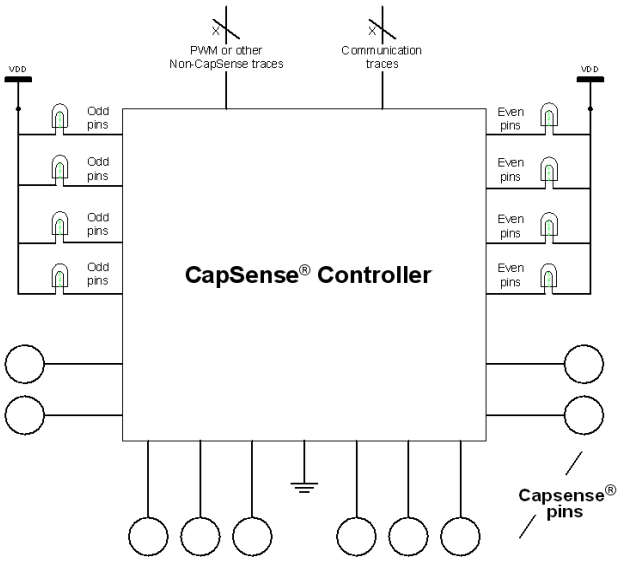


5.6 ピンの割り当て

CapSense センサー配線と通信配線と非 CapSense 配線間の相互影響を制限する効果的な方法としては、それぞれをポートの割り当てで隔離することです。図 5-2 に、32 ピン QFN パッケージ向けの分離の基本バージョンを示します。各機能を分離しているため、CapSense コントローラーは、通信、LED、および検知トレースが交差しないように方向を設定されます。

多数の VSS ピンのあるデバイスでは、すべての VSS ピンを共通の GND 面に集積する必要があります。

図 5-2. 推奨: 通信、CapSense と LED のポートによる隔離



CapSense コントローラーのアーキテクチャは、偶数と奇数のポート ピン番号で、電流の割当量を制限します。CapSense コントローラーでは、奇数ポート ピンの電流の割当量が 100mA であり、すべての奇数ポート ピンを通る電流合計は 100mA を超過してはなりません。合計電流割当量の制限に加えて、各ポート ピンには最大電流限度があり、これは「CapSense コントローラー データシート」で定義します。

すべての CapSense コントローラーは、高電流シンクとソース対応のポート ピンを備えています。高電流シンクやソースのポート ピンを使用する場合、デバイスのグランドピンに最も近いポートを使用し、ノイズを最小限に抑える必要があります。

静電容量検知アプリケーションの検知回線は、センサーのみに接続する必要があります。ISSP プログラミング ヘッダなどその他のボード要素に取り付けた検知回線は、外部ノイズにより敏感であり、伝導経路の表面面積が大きくなることでより多くの寄生容量が生じます。プログラム ピン、P1[0]と P1[1]にセンサーを配置することは避けてください。

また、プログラミング ピン P1[0]と P1[1]は、他の I/O ピンに比べて POR または XRES イベントへの応答が異なっています。XRES イベントのあと、どちらのピンも、HI-Z 駆動モードに達する前に抵抗ゼロ 駆動モードに入って、接地まで電圧が低下します。POR イベントのあと、P1[0]は 1 を出力してからしばらくの間抵抗ゼロ状態に入り、最終的に HI-Z 駆動モード状態に達します。POR のあと、P1[1]はしばらく抵抗ゼロ状態に入ってから、HI-Z 駆動モードに移行します。これらのピンをアプリケーションで使用する際には、この点に留意してください。

5.7 プリント基板レイアウトのガイドライン

詳細なプリント基板レイアウトのガイドラインについては、「CapSense Design 入門」を参照してください。

6. リソース

6.1 ウェブサイト

サイプレス CapSense コントローラー ウェブサイトでは、この節で討論した参照資料すべてを見ることができます。

CY8C20x34 ウェブページには、CapSense CY8C20x34 デバイス ファミリのさまざまな技術リソースがあります。

6.2 データシート

CapSense CY8C20x34 デバイス ファミリのデータシートは、www.cypress.com で入手することができます。

- CY8C20234、CY8C20334、CY8C20434、CY8C20534、CY8C20634
- CY8C20224、CY8C20324、CY8C20424、CY8C20524

6.3 テクニカル リファレンス マニュアル

サイプレスは以下の技術マニュアルを作成しており、トップレベルのアーキテクチャ図や抵抗器、およびタイミング図など、CapSense コントローラーの機能にすばやく簡単にアクセスすることができます。

- PSoC® CY8C20x24、CY8C20x34 テクニカル リファレンス マニュアル (TRM)

6.4 開発キット

6.4.1 ユニバーサル CapSense コントローラー - 基本キット 1

CY3280-BK1 ユニバーサル CapSense コントローラー キットは、事前定義された制御回路とプラグイン ハードウェアを使用して、容易に CapSense デザインのプロトタイプを作成し、デバッグできるように設計されています。

6.4.2 ユニバーサル CapSense モジュール基板

6.4.2.1 簡単なボタン モジュール基板

CY3280-BSM シンプル ボタン モジュールは、それぞれ 10 個の CapSense ボタンと LED で構成されています。このモジュールはあらゆる CY3280 ユニバーサル CapSense コントローラーボードと接続します。

6.4.2.2 マトリックス ボタン モジュール基板

CY3280-BMM マトリックス ボタン モジュールは、4×4 マトリックス形式として構成される 8 個の LED と 8 個の CapSense センサーからなります (すなわち、物理的ボタン 16 個が形成される)。このモジュールはあらゆる CY3280 ユニバーサル CapSense コントローラーボードと接続します。

6.4.2.3 リニア スライダー モジュール基板

CY3280-SLM リニア スライダー モジュールは 5 個の CapSense ボタン、1 個のリニア スライダー (10 個のセンサー付) および 5 個の LED から成ります。このモジュールはあらゆる CY3280 ユニバーサル CapSense コントローラー ボードと接続します。

6.4.2.4 ラジアル スライダー モジュール基板

CY3280-SRM ラジアル スライダー モジュールは 4 個の CapSense ボタン、1 個のラジアル スライダー (10 個のセンサー付) および 4 個の LED から成ります。このモジュールはあらゆる CY3280 ユニバーサル CapSense コントローラー ボードと接続します。

6.4.2.5 ユニバーサル CapSense プロトタイプモジュール

CY3280-BBM ユニバーサル CapSense プロトタイプモジュールで、付属コントローラーボード上の 44 ピン コネクタに接続されたすべての信号へのアクセスが可能です。プロトタイプ モジュール ボードはユニバーサル CapSense コントローラーボードと併用し、その他の専用のユニバーサル CapSense コントローラーボードにはない追加機能を実行することができます。

6.4.3 インサーキット エミュレーション (ICE) キット

ICE ポッドは、フレックス ケーブルを通してパッケージ専用ポッドを使用し、プロトタイプ システムまたはプリント基板で CY3215-DK インサーキット エミュレータとターゲットの PSoC デバイスの相互接続を可能にします。次のポッドを使用することができます。

- QFN CY8C20334 PSoC デバイス デバッグ用 CY3250-20334QFN インサーキット エミュレーション(ICE) ポッド(2)
- QFN CY8C20434 PSoC デバイス デバッグ用 CY3250-20434QFN インサーキット エミュレーション(ICE) ポッド(2)

6.5 サンプル ボード ファイル

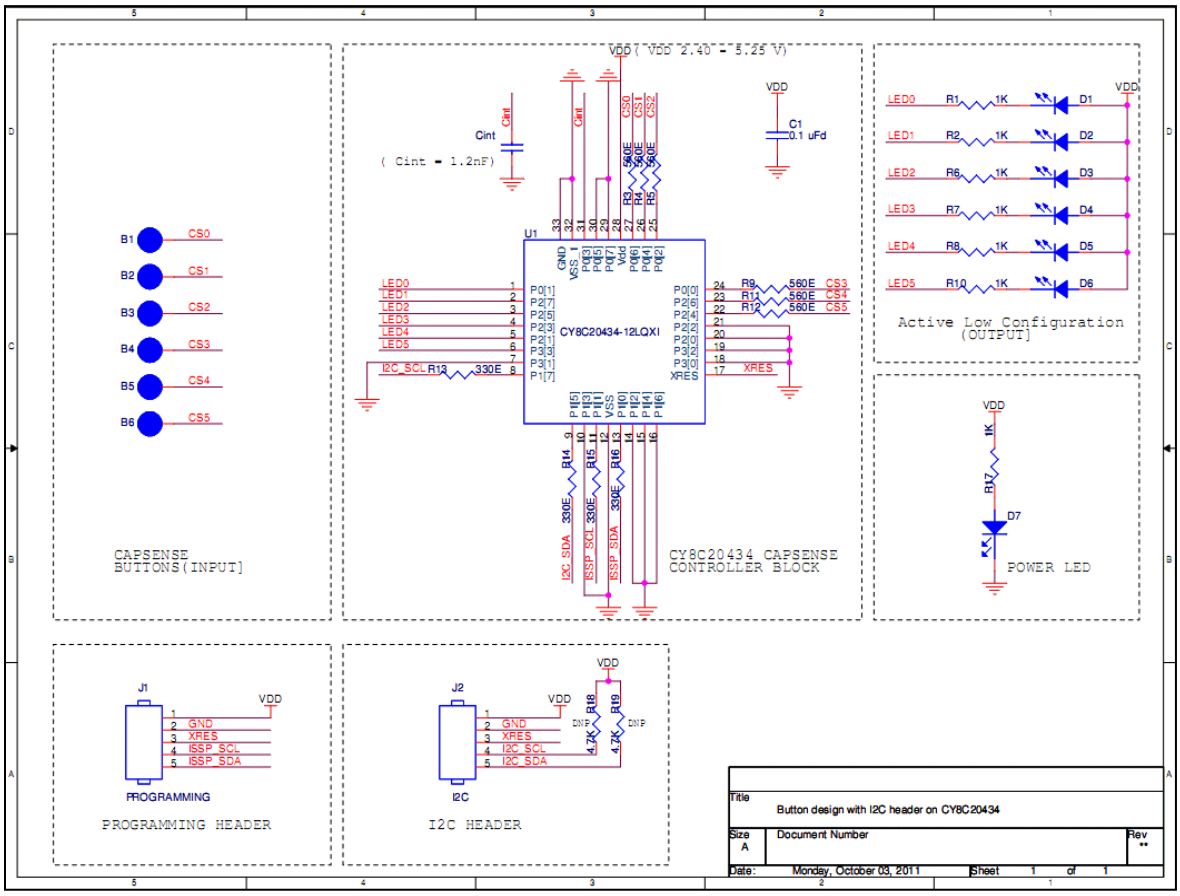
サイプレスは、参照として使用してプリント基板設計プロセスをすばやく完了できる回路図およびボード ファイルのサンプルを提供しています。

6.5.1 CY8C20434 の I²C ヘッダのあるボタンデザイン

図 6-1 は、CY8C20434 の I²C ヘッダのあるボタンデザイン回路図を示しています。デザインは以下をサポートします:

- 6 個の CapSense センサー。センサーは、CY8C20434-12LQXI デバイスのピン P0[0]、P0[2]、P0[4]、P0[6]、P2[4]、および P2[6]に割り当てられる
- CY8C20434-12LQXI のピン P0[1]、P2[1]、P2[3]、P2[5]、P2[7]、および P3[3]に接続され、D1、D2、D3、D4、D5、および D6 の LED を駆動している 6 個の GPO
- プログラミング ヘッダ J1 による CY8C20434-12LQXI のプログラミング
- I²C ヘッダ J2 による CY8C20434-12LQXI との I²C 通信

図 6-1. CY8C20434 の I²C ヘッダのあるボタンデザイン - ボード回路図

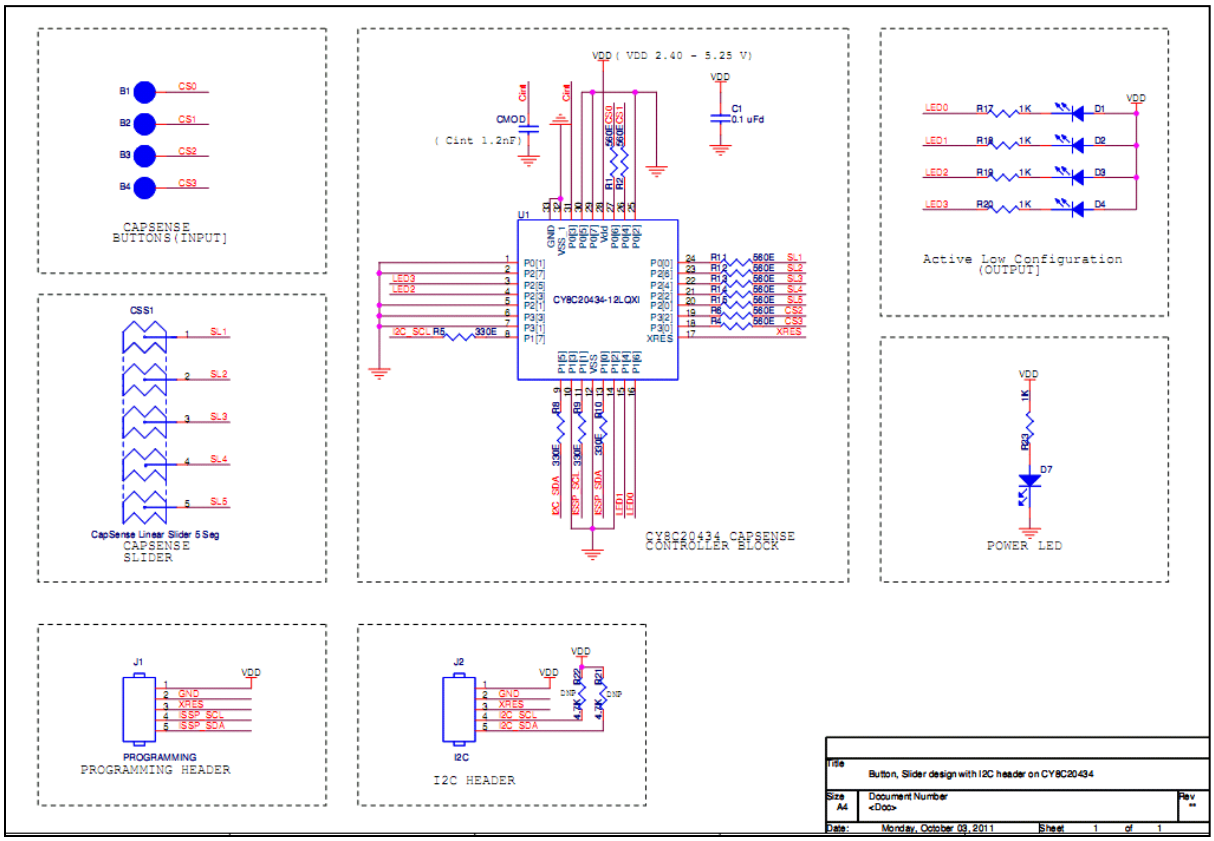


6.5.2 CY8C20434 の I²C ヘッダのあるボタンおよびスライダーのデザイン

図 6-2 は、CY8C20434 の I²C ヘッダのあるボタンとスライダーのデザイン回路図を示しています。このデザインは以下をサポートします:

- 4 個の CapSense センサー。センサーは、CY8C20434-12LQXI デバイスのピン P0[4]、P0[6]、P3[0]、および P3[2]に割り当てられる
- 5 個のセグメントを持つリニアスライダー。セグメントは、CY8C20434-12LQXI デバイスのピン P0[0]、P2[0]、P2[2]、P2[4]、および P2[6]に割り当てられる
- D1、D2、D3、および D4 の LED を駆動する、CY8C20434-12LQXI のピン P1[4]、P1[6]、P2[5]、および P2[7]に接続された 4 個の GPO
- プログラミング ヘッダ J1 による CY8C20434-12LQXI のプログラミング
- I²C ヘッダ J2 による I²C と CY8C20434-12LQXI との通信

図 6-2. CY8C20434 に I²C ヘッダをもつボタンおよびスライダーの設計



6.6 PSoC Designer

サイプレスは、専用の統合設計環境である PSoC Designer を提供しています。PSoC Designer により、アナログおよびデジタル ブロックを設定し、ファームウェアを開発し、設計をチューニングすることができます。CapSense を含めた、あらかじめ特性確定済みのアナログおよびデジタル機能のライブラリを使い、ドラッグ & ドロップによる設計環境でアプリケーションを開発できます。PSoC Designer には内蔵の C コンパイラと組み込みプログラマが含まれています。複雑なデザイン用には pro コンパイラがあります。

6.7 PSoC Programmer

PSoC Programmer は PSoC デバイスをプログラムするための柔軟性のある統合プログラム アプリケーションです。PSoC Programmer は PSoC Designer および PSoC Creator と併用し、PSoC デバイスにデザインをプログラムします。

PSoC Programmer は、API をもつハードウェア レイヤを、プログラマとブリッジ デバイスを使用する設計専用アプリケーションに提供します。PSoC Programmer ハードウェア層は、COM ガイド ドキュメントで説明されています。また、コード例も C#、C、Perl、および Python の言語で提供されます。

6.8 CapSense データ表示ツール

CapSense 設計中に、チューニングとデバッグを目的として、関連する CapSense データ (raw カウント、ベースライン、差分カウントなど) を何度も監視したい場合があります。

アプリケーション ノート「AN2397 – CapSense データ表示ツール」は、CapSense データの表示とロギング用の適切なツールを特定・使用するために役立つ必要な情報を提供します。

6.9 コード例

サイプレスは、設計を実現して高速で実行するコード例をいくつか提供しています。CapSense 開発キットでのさまざまな CapSense の機能の評価については、「CapSense コントローラー コード例のデザイン ガイド」を参照してください。

6.10 デザイン サポート

サイプレスには多くの設計サポート チャンネルがあり、CapSense ソリューションの成功を期しています。

- 知識ベース記事 – 製品ファミリ別技術記事を参照したり、さまざまな CapSense のトピックの検索が行なえます。
- CapSense アプリケーション ノート – この文書で記載された情報に基づく広範囲なアプリケーション ノート
- ホワイト ペーパー: 静電容量タッチ インターフェースの高度なトピックについて学びます。
- サイプレス開発コミュニティ: サイプレス技術コミュニティに参加し、情報交換できます。
- CapSense 製品セレクタ ガイド – サイプレス CapSense 製品ラインの製品提供の全体を見ることができます。
- 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。
- 品質と信頼性 – 当社の品質ウェブサイトで信頼性および製品認定報告書をご覧ください。サイプレスは、完全な顧客満足度の達成に取り組んでいます。
- 技術サポート: 一流の技術サポートをオンラインで提供しています。

用語集

AMUXBUS

PSoC 内部で利用可能なアナログマルチプレクサバスで、I/O ピンを複数の内部アナログ信号に接続するのに役立ちます。

SmartSense™ Auto-Tuning (SmartSense 自動チューニング)

設計段階後に最適な性能を得るためにセンシングパラメータを自動的に設定し、システム，製造，および環境の変化を継続的に補正する CapSense アルゴリズム。

Baseline (ベースライン)

センサー上に人の指が存在しない場合に Raw カウントの傾向を推定するファームウェアアルゴリズムから生じる値。ベースラインは、Raw カウントの急激な変化に対する感度が低く、差分カウントを計算するための基準点を提供します。

Button または Button Widget

センサーのアクティブ状態または非アクティブ状態 (つまり 2 つの状態のみ) を報告できる、関連付けられたセンサーを持つウィジェット。たとえば、センサー上の指の接触または非接触状態を検出できます。

Difference Count (差分カウント)

Raw カウントと Baseline の違い。差が負の場合、またはそれがノイズしきい値を下回る場合、Difference Count は常にゼロに設定されます。

Capacitive Sensor (静電容量センサー)

A conductor and substrate, such as a copper button on a printed circuit board (PCB), which reacts to a touch or an approaching object with a change in capacitance.

静電容量の変化を伴って接触または接近してくる物体に反応する、プリント基板 (PCB) 上の銅ボタンなどの導体および基板。

CapSense®

サイプレスのタッチセンシングユーザーインターフェースソリューション。業界第 1 位のソリューションで、第 2 位の 4 倍の売上を誇る。

CapSense Mechanical Button Replacement (MBR)

メカニカルボタンを静電容量式ボタンにアップグレードするためのサイプレスのソリューションで、センサーパラメータを構成するための最小限の技術的努力しか必要とせず、ファームウェア開発を必要としません。これらのデバイスには、CY8CMBR3XXX および CY8CMBR2XXX ファミリが含まれます。

Centroid または Centroid Position

スライダーの解像度で指定された範囲内でスライダー上の指の位置を示す数値。この数値は CapSense Centroid 計算アルゴリズムによって計算されます。

Compensation IDAC (補正 IDAC)

CSD が過剰なセンサーC_Pを補正するために使用されるプログラム可能な定電流源。この IDAC は、変調 IDAC とは異なり、CSD ブロック内のシグマデルタ変調器によって制御されません。

CSD (CapSense シグマデルタ**)**

CapSense シグマデルタ(CSD)は、静電容量センシングアプリケーション用に自己容量 (self-cap とも呼ばれる) 測定を実行するサイプレスの特許取得済みの方法です。

CSD モードでは、検知システムは電極の自己容量を測定し、自己容量の変化を検出して指の有無を識別します。

Debounce

接触が有効になるために接触が存在する連続スキャンサンプルの数を定義するパラメーター。このパラメーターは、誤った接触信号を排除するのに役立ちます。

指の接触は、スキャンサンプルの連続する Debounce 数に対して、差分カウントが指のしきい値+ヒステリシスより大きい場合にのみ報告されます。

Driven-Shield (駆動シールド)

CSD が耐液性を実現するために使用される技術で、シールド電極は位相と振幅がセンサー切り替え信号と等しい信号で駆動されます。

Electrode (電極)

PCB, ITO, または FPCB 上のパッドまたは層の導電性材料。電極は CapSense デバイスのポートピンに接続され、CapSense センサーとして、または CapSense 機能に関連する特定の信号を駆動するために使用されます。

Finger Threshold (指しきい値**)**

センサーの状態を判断するためにヒステリシスとともに使用されるパラメーター。差カウントが指しきい値+ヒステリシスよりも高い場合はセンサー状態が ON と報告され、差カウントが指しきい値 - ヒステリシスを下回る場合はセンサー状態が OFF と報告されます。

Ganged Sensors (連動センサー)

複数のセンサーを接続し、それらを単一のセンサーとしてスキャンする方法。近接センシングのためのセンサー面積を増やし、消費電力を減らすために使用されます。
システムが低電力モードのときに電力を削減するには、すべてのセンサーを個別にスキャンするのではなく、すべてのセンサーをまとめて 1 つのセンサーとしてスキャンする時間を短縮できます。ユーザーがいずれかのセンサーに触れると、システムはアクティブモードに移行して、アクティブになっている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 はファームウェアのセンサー連動に対応しています。つまり、スキャンのために複数のセンサーを同時に AMUXBUS に接続できます。

Gesture (ジェスチャ)

ジェスチャは、スワイプやピンチズームなどの、ユーザーが実行するアクションです。CapSense には、定義済みの接触パターンに基づいてさまざまなジェスチャを識別するジェスチャ検出機能があります。CapSense コンポーネントでは、ジェスチャ機能はタッチパッドウィジェットでのみ対応しています。

Guard Sensor (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囲む銅トレース。液体の流れを検出するために使用されます。ガードセンサーが作動すると、ファームウェアは他のすべてのセンサーのスキャンを無効にして誤った接触を防ぐことができます。

Hatch Fill, Hatch Ground または Hatched Ground

静電容量式センシング用に PCB を設計する際は、ノイズ耐性を高めるために、接地された銅平板をセンサーの周囲に配置する必要があります。しかし、固体グランドはセンサーの寄生容量を増加させ、これは望ましくありません。したがって、グランドは特別な Hatch パターンで埋められるべきです。Hatch パターンには、密接に配置されたメッシュのような交差した線を持ち、線の幅と 2 本の線の間隔によって塗りつぶしの割合が決まります。耐液性の場合、シールド電極と呼ばれるこの Hatch Fill は、グランドの代わりにシールド信号で駆動されます。

Hysteresis (ヒステリシス)

センサーの状態を判断するために Finger Threshold と組み合わせて使用される、システムノイズによるセンサー状態出力のランダムトグルを防ぐために使用されるパラメーター。Finger Threshold を参照してください。

IDAC (Current-Output Digital-to-Analog Converter)

PSoC 内部で利用可能な CapSense および ADC 動作に使用されるプログラム可能な定電流源。

Liquid Tolerance (耐液性)

液滴，流動液体，またはミストの存在下で確実に機能する静電容量式センシングシステムの機能。

Linear Slider (リニアスライダー)

指の物理的な位置 (単軸) を検出するために、特定の直線状に配置された複数のセンサーからなるウィジェット。

Low Baseline Reset (低ベースラインリセット)

Raw Count が Negative Noise Threshold 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。低ベースラインリセット値を超えると、ベースラインは現在の Raw カウントにリセットされます。

Manual-Tuning (手動チューニング)

CapSense パラメーターを設定 (または調整) する手動プロセス。

Matrix Buttons (マトリックスボタン)

縦横に配置されたセンサーの交差部分での人間の指の存在 (タッチ) の有無を検出するために使用される、マトリックス状に配置された 2 つ以上のセンサーからなるウィジェット。
横軸のセンサー数を M、縦軸のセンサー数を N とすると、マトリックスボタンウィジェットは M + N ポートピンのみを使用して M×N 個の交差点を監視できます。
CSD センシング方式 (自己容量) を使用している場合、このウィジェットは一度に 1 つの交差位置のみで有効な接触を検出できます。

Modulation Capacitor (CMOD, 変調コンデンサ)

自己容量検出モードで CSD ブロックを動作させるために必要な外部コンデンサ。

Modulator Clock (変調器クロック)

センサースキャン中に CSD ブロックからの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。このクロックは Raw カウントカウンターにも供給されます。スキャン時間 (前処理時間と後処理時間を除く) は $(2^N - 1)$ /変調器クロック周波数 で与えられます。ここで、N はスキャン解像度です。

Modulation IDAC (変調 IDAC)

変調 IDAC はプログラム可能な定電流源で、その出力は CSD ブロック内のシグマデルタ変調器出力によって制御され (ON/OFF)、AMUXBUS 電圧を V_{REF} に維持します。この IDAC によって供給される平均電流は、センサコンデンサによって引き出される平均電流に等しくなります。

Mutual-Capacitance (相互静電容量)

他の電極 (例えば RX) に対する電極 (例えば TX) に関連する静電容量は相互静電容量として知られている。

Negative Noise Threshold (負のノイズしきい値)

負の方向に現れるスプリアス信号と通常のノイズを区別するために使用されるしきい値。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組み合わせて使用されます。

Raw カウントが負のノイズしきい値内にある限り、ベースラインは Raw カウントの変化を追跡するように更新されます。つまり、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より小さくなります。

このようなスプリアス信号を負の方向に向かわせる可能性のある状況には、次のものがあります。電源投入時のセンサー上の指，センサーの近くに置かれた金属物体の取り外し，耐液性 CapSense 対応製品の水からの取り外し，およびその他の突然の環境変化。

Noise (CapSense ノイズ)

センサーが OFF 状態 (接触なし) にあるときの Raw カウントの変動。ピークツーピークカウントとして測定されます。

Noise Threshold (ノイズしきい値)

センサーの信号とノイズを区別するためのパラメーター。Raw カウント – ベースラインがノイズしきい値より大きい場合は、有効な信号である可能性があります。差がノイズしきい値より小さい場合、Raw カウントにはノイズ以外何も含まれません。

Overlay (オーバーレイ)

容量センサーを覆い、接触面として機能するプラスチックやガラスなどの非導電性材料。センサー付きの PCB は、直接オーバーレイの下に配置されるか、バネを介して接続されます。製品の覆いは、しばしばオーバーレイになります。

Parasitic Capacitance (C_P, 寄生容量)

寄生容量は、PCB トレース，センサーパッド，ビア，およびエアギャップによって発生するセンサー電極の固有容量です。CSD の感度が低下するため、望ましくありません。

Proximity Sensor (近接センサー)

物理的な接触なしで近くの物体の存在を検出できるセンサー。

Radial Slider (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に配置された複数のセンサーからなるウィジェット。

Raw Count (Raw カウント)

センサーの物理的な静電容量を表す CapSense ハードウェアブロックの未処理のデジタルカウント出力。

Refresh Interval (更新間隔)

センサーの 2 つの連続するスキャンの間の時間。

Scan Resolution (スキャン解像度)

CSD ブロックによって生成された Raw カウントの分解能 (ビット単位)。

Scan Time (スキャン時間)

センサーのスキャンが完了するのにかかる時間。

Self-Capacitance (自己容量)

回路グランドに対する電極に関連する静電容量。

Sensitivity (感度)

Raw カウントの変化はセンサー容量の変化に対応し、カウント/pF で表されます。センサーの感度は、基板レイアウト，オーバーレイ特性，検出方法，および調整パラメーターによって異なります。

Sense Clock (センスクロック)

CSD 検出方式用のスイッチトキャパシタ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。

Sensor

Capacitive Sensor を参照してください。

Sensor Auto Reset (センサー自動リセット)

システム障害のため、または金属製の物体がセンサーの近くに連続的に存在する場合に、センサーが誤って接触状態を報告しないようにするための設定。
センサー自動リセットが有効になっていると、差分カウントがノイズしきい値よりも大きい場合でも、ベースラインは常に更新されます。これによりセンサーが ON 状態を無期限に報告することを防ぎます。センサー自動リセットが無効になっていると、差分カウントがノイズしきい値より小さい場合にのみベースラインが更新されます。

Sensor Ganging

Ganged Sensors を参照してください。

Shield Electrode (シールド電極)

水や他の液体の存在による誤った接触を防ぐため、センサーの周りを銅で埋めます。シールド電極は、CSD ブロックから出力されるシールド信号によって駆動されます。Driven-Shield を参照してください。

Shield Tank Capacitor (C_{SH})

寄生容量が大きい大きなシールド層がある場合に、CSD シールドの駆動能力を高め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)。

Signal (CapSense シグナル)

差分カウントは Signal とも呼ばれます。Difference Count を参照してください。

Signal-to-Noise Ratio (SNR, 信号対雑音比)

接触時のセンサー信号と非接触時センサーの雑音信号との比率。

Slider Resolution (スライダー解像度)

スライダー上で分けられる指の位置の総数を示すパラメーター。

Touchpad (タッチパッド)

接触の X 位置と Y 位置を検出するために特定の水平方向と垂直方向に配置された複数のセンサーで構成されるウィジェット。

Trackpad (トラックパッド)

Touchpad を参照してください。

Tuning (調整)

CapSense 操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見つけるプロセス。

V_{REF}

PSoC 内部で利用可能な CapSense および ADC 動作に使用されるプログラム可能な基準電圧ブロック。

Widget (ウィジェット)

1 つのセンサーまたは類似のセンサーの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。対応しているウィジェットは、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です。

改訂履歴

改訂履歴

文書名: AN66269 - CY8C20x34 CapSense®デザイン ガイド 文書番号: 001-79091		
版	**発効日**	**変更内容**
**	05/17/2012	これは英語版 001-66269 Rev. *D を翻訳した日本語版 001-79091 Rev. **です。
*A	06/08/2015	これは英語版 001-66269 Rev. *G を翻訳した日本語版 001-79091 Rev. *A です。
*B	07/23/2019	これは英語版 001-66269 Rev. *J を翻訳した日本語版 001-79091 Rev. *B です。