サイプレスはインフィニオン テクノロジーズになりました

この表紙に続く文書には「サイプレス」と表記されていますが、これは同社が最初にこの製品を開発したからです。新規および既存のお客様いずれに対しても、引き続きインフィニオンがラインアップの一部として当該製品をご提供いたします。

文書の内容の継続性

下記製品がインフィニオンの製品ラインアップの一部として提供されたとしても、それを理由としてこの文書に変更が加わることはありません。今後も適宜改訂は行いますが、変更があった場合は文書の履歴ページでお知らせします。

注文時の部品番号の継続性

インフィニオンは既存の部品番号を引き続きサポートします。ご注文の際は、データシート記載の注文部品番号をこれまで通りご利用下さい。

CapSense® コントローラー サンプルコード

文書番号: 001-85884 Rev. *C

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

© Cypress Semiconductor Corporation, 2012-2019. 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, CapsSense, 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

はじめに 8

概要 8

サンプル コード 1　SmartSense 付きボタン スライダー (CY8C20xx6A CapSense コントローラー) 11

1.1 プロジェクト名 11
1.2 概要 11
1.3 ハードウェアのセットアップ 11
1.3.1 要求事項 11
1.3.2 組み立て 11
1.3.3 ボードの設定 12
1.4 回路図 13
1.5 ソフトウェアの準備 13
1.5.1 ツール 13
1.5.2 ユーザーモジュール リスト 14
1.5.3 ユーザーモジュール パラメーター、グローバル リソース 14
1.6 動作 14
1.7 サンプルコードの実行 15
1.8 I²C を経由した CapSense データの読み出し 15
1.8.1 Bridge Control Panel ツールの起動 15
1.8.2 BTN0 の raw カウント、ベースライン、差分カウントの読み出し 16
1.8.3 SLD0 の raw カウント、ベースライン、差分カウント、スライダー ポジションの読み出し 16

サンプル コード 2　I²C によるホスト通信(CY8C20xx6A CapSense コントローラー使用) 17

2.1 プロジェクト名 17
2.2 概要 17
2.3 ハードウェアのセットアップ 17
2.3.1 要求事項 17
2.3.2 アセンブリ 17
2.3.3 ボードの設定 18
2.4 回路図 18
2.5 ソフトウェアの準備 19
2.5.1 ツール 19
2.5.2 ユーザーモジュール リスト 19
2.5.3 ユーザーモジュール パラメーター、グローバル リソース 19

2.6 動作 19
2.7 サンプルコードの実行 20
2.8 I²C を経由した CapSense データの読み出し 20
2.8.1 Bridge Control Panel の起動 20
2.8.2 BTN0 の raw カウント、ベースライン、差分カウントの読み出し 21
2.8.3 SLD0 の raw カウント、ベースライン、差分カウント、スライダー ポジションの読み出し 21

サンプル コード 3 UART によるデータ送信 (CY8C20xx6A CapSense コントローラー使用) 22

3.1 プロジェクト名 22
3.2 概要 22
3.3 ハードウェアの準備 22
3.3.1 要求事項 22
3.3.2 アセンブリ 22
3.3.3 ボードの設定 23
3.4 回路図 24
3.5 ソフトウェアの準備 25
3.5.1 ツール 25
3.5.2 ユーザーモジュールの一覧 25
3.5.3 ユーザーモジュール パラメーター、グローバル リソース 25
3.6 動作 25
3.7 コード サンプルの実行 26
3.8 CapSense データを MultiChart でプロットする 27

サンプル コード 4 絶対センサー静電容量の測定 (CY8C20xx6A CapSense コントローラー使用) 29

4.1 プロジェクト名 29
4.2 概要 29
4.3 ハードウェアの準備 29
4.3.1 要求事項 29
4.3.2 アセンブリ 29
4.3.3 ボードの設定 30
4.4 回路図 31
4.5 ソフトウェアの準備 31
4.5.1 ツール 31
4.5.2 ユーザーモジュール 32
4.5.3 ユーザーモジュール パラメーター、グローバル リソース 32
4.6 動作 32
4.7 サンプルコードの実行 34

サンプル コード 5 絶対センサー静電容量の測定 (CY8C21x34/B CapSense コントローラー使用) 38

5.1 プロジェクト名 38
5.2 概要 38
5.3 ハードウェアの準備 38
5.3.1 要求事項 38
5.3.2 アセンブリ 38
5.3.3 ボードの設定 39

5.4 回路図 40
5.5 ソフトウェアの準備 40
5.5.1 ツール 40
5.5.2 ユーザーモジュール 41
5.5.3 ユーザーモジュール パラメーター、グローバル リソース 41
5.6 動作 41
5.7 コードの実行 43

サンプル コード 6 絶対センサー静電容量の測定 (CY8C20x34 CapSense コントローラー使用) 47

6.1 プロジェクト名 47
6.2 概要 47
6.3 ハードウェアの準備 47
6.3.1 要求事項 47
6.3.2 組み立て 47
6.3.3 ボードの設定 48
6.4 回路図 49
6.5 ソフトウェアの準備 50
6.5.1 ツール 50
6.5.2 ユーザーモジュール 50
6.5.3 ユーザーモジュール パラメーター、グローバル リソース 50
6.6 動作 50
6.7 サンプルコードの実行 51

サンプル コード 7 CSD の調整 (CY8C20xx6A CapSense コントローラー使用) 54

7.1 プロジェクト名 54
7.2 概要 54
7.3 ハードウェアのセットアップ 54
7.3.1 要求事項 54
7.3.2 組み立て 54
7.3.3 ボードの設定 55
7.4 回路図 55
7.5 ソフトウェアの準備 56
7.5.1 ツール 56
7.5.2 ユーザーモジュールと配置 56
7.5.3 ユーザーモジュール パラメーター、グローバル リソース 56
7.6 動作 56
7.7 コードの実行 57
7.8 I^2C を経由した CapSense データの読み出し 57
7.8.1 Bridge Control Panel ツールの起動 58
7.8.2 BTN0 の raw カウント、ベースライン、差分カウントの読み出し 58
7.8.3 SLD0 の raw カウント、ベースライン、差分カウント、スライダー ポジションの読み出し 59
7.9 CSD ユーザーモジュールの調整 59
7.9.1 CSD ユーザーモジュールの設定を行う 60

サンプル コード 8　フィードバック抵抗付き CSD の調整 (CY8C21x34/B CapSense コントローラー使用) 63

8.1 プロジェクト名 63
8.2 概要 63
8.3 ハードウェアのセットアップ 63
8.3.1 要求事項 63
8.3.2 アセンブリ 63
8.3.3 ボードの設定 64
8.4 回路図 65
8.5 ソフトウェアの準備 65
8.5.1 ツール 65
8.5.2 ユーザーモジュールと配置 66
8.5.3 ユーザーモジュール パラメーター、グローバル リソース 66
8.6 動作 66
8.7 コードの実行 67
8.8 I²C を経由した CapSense データの読み出し 67
8.8.1 Bridge Control Panel ツールの起動 67
8.8.2 BTN0 の raw カウント、ベースライン、差分カウントの読み出し 68
8.8.3 SLD0 の raw カウント、ベースライン、差分カウント、スライダー ポジションの読み出し 68
8.9 CSD ユーザーモジュールの調整 69
8.9.1 CSD UM の設定を行う 69

サンプル コード 9　CSA_EMC の調整 (CY8C20x34 CapSense コントローラー使用) 72

9.1 プロジェクト名 72
9.2 概要 72
9.3 ハードウェアの準備 72
9.3.1 関連ハードウェア 72
9.3.2 アセンブリ 72
9.3.3 ボードの設定 73
9.4 回路図 73
9.5 ソフトウェアの準備 74
9.5.1 ツール 74
9.5.2 ユーザーモジュール 74
9.5.3 ユーザーモジュール パラメーター、グローバルリソース 74
9.6 動作 74
9.7 コードの実行 75
9.8 I²C を経由した CapSense データの読み出し 75
9.8.1 BTN0 のローカウント、ベースライン、差分カウントの読み出し 76
9.8.2 SLD0 の raw カウント、ベースライン、差分カウント、スライダー ポジションの読み出し 76
9.9 CSA_EMC ユーザーモジュールの調整 77
9.9.1 CSD ユーザーモジュールの設定を行う 77

サンプル コード 10　CY8C20xx6A の SmartSense による CapSense ボタンの消費電力の最適化 80

10.1 プロジェクト名 80
10.2 概要 80

10.3 ハードウェアの準備 80
10.3.1 関連ハードウェア 80
10.3.2 組み立て 80
10.4 ボードの設定 80
10.5 回路図 81
10.6 ソフトウェアの準備 82
10.6.1 ツール 82
10.6.2 ユーザーモジュール 82
10.6.3 ユーザーモジュール パラメーター、グローバル リソース 82
10.7 動作 82
10.8 コードの実行 83

付録 85

CSD ユーザーモジュールの設定 85
他の CapSense デバイスへのサンプルコードの移行 90
絶対センサー静電容量測定 91
関連文書 91
設計ガイド 91
データシート 91
アプリケーションノート 91
キット ガイド 92
略語 92
改訂履歴 93

はじめに

Cypress CapSense® ソリューションは、お客様のデザインにエレガントで、信頼性が高く、使い易い静電容量タッチセンシング機能をもたらしています。当社の静電容量タッチセンシング ソリューションは、40 億個以上の機械式ボタンを置き換えてきました。CapSense により、携帯電話、PC、民生電子機器および大型家電製品のデザインの外観が変わりました。サイプレスの堅牢な CapSense ソリューションは、当社のフレキシブルなプログラマブル システム オン チップ (PSoC®) アーキテクチャを活用しています。これにより、製品化までの時間を短縮し、重要なシステム機能を統合し、材料費を削減します。

概要

本書では、お客様の実験を支援し、サイプレスの CapSense コントローラー、CapSense キット、CapSense ユーザーモジュールの機能を評価するために、10 個の CapSense サンプルコードの概要を記載しています。また、I^2C および UART インターフェースを介してセンサーデータを読み出し、プロットする方法についても説明します。この文書を読むには、CapSense のセンシング技術を知っている必要があります。CapSense 全般の原理と動作の詳細については、CapSense 入門を参照してください。

注: 本アプリケーションノートのサンプルコードは、本書で参照するキットに固有のものです。カスタムボードにサンプルコードを移植するには、CapSense 入門で主要な設計留意点やレイアウトのベストプラクティスを参照してください。本書に、必要なことが書かれていなかった、疑問に対する答がなかった、またはご意見がある場合は、capsense@cypress.com へ e メールでご連絡ください。

評価を容易にするために、サンプルコードは簡単なものから複雑なものへ順番に提示します。

- **評価にあたって**: 初めての場合、CapSense タッチセンサー機能のデモから開始することをお勧めします。サンプルコード 1: SmartSense 付きボタン スライダー (CY8C20xx6A CapSense コントローラー) は、SmartSense 自動チューニング静電容量センサアルゴリズムを使用したタッチボタンやスライダーが、いかに簡単で迅速に実装できるかを示しています。
- **通信方法**: CapSense の仕組みを理解したら、次にデータの送受信を CapSense コントローラーで行う方法を学習します。この通信は、CapSense センサーの調整や、CapSense コントローラーとホストプロセッサをつなぐために必要です。サンプルコード 2: I2C によるホスト通信(CY8C20xx6A CapSense コントローラー使用)およびサンプルコード 3: UART によるデータ送信 (CY8C20xx6A CapSense コントローラー使用) では、CapSense コントローラーで通信する方法について詳しく説明しています。
- **調整**: 次のステップでは、安定した動作のために CapSense センサーを調整します。調整手順を完了するには、まず PCB 上の静電容量センサーの寄生容量について理解しなければなりません。サンプルコード 4:絶対センサー静電容量の測定 (CY8C20xx6A CapSense コントローラー使用)、サンプルコード 5: 絶対センサー静電容量の測定 (CY8C21x34/B CapSense コントローラー使用)、およびサンプルコード 6: 絶対センサー静電容量の測定 (CY8C20x34 CapSense コントローラー使用)は、様々な CapSense デバイスの絶対センサー静電容量を測定します。サンプルコード 7: CSD の調整 (CY8C20xx6A CapSense コントローラー使用)、サンプルコード 8: フィードバック抵抗付き CSD の調整 (CY8C21x34/B CapSense コントローラー使用)、およびサンプルコード 9: CSA_EMC の調整 (CY8C20x34 CapSense コントローラー使用) は、CapSense コントローラーの調整手順について説明します。
- **消費電力に応じて最適化**: 最後、サンプルコード 10: CY8C20xx6A の SmartSense による CapSense ボタンの消費電力の最適化では、エンドアプリケーションで電池寿命を伸ばすために、CapSense コントローラーの消費電力を最適化する方法について説明しています。

注: この設計ガイドの適用対象は、CY8C20xx6A、CY8C20x34、CY8C21x34/B、および CY8C24x94 の 4 つの CapSense デバイスです。サンプルコードを他の CapSense デバイスに移植する方法については、本書の付録を参照してください。付録では、CSD ユーザーモジュールの設定に関する命令も説明しています。サンプルコードの作成や修正には CSD ユーザーモジュールの設定が必要です。

表 1. サンプルコード一覧

サンプルコード	サンプルコード	対象キット	説明
1	SmartSense 付きボタン スライダー (CY8C20xx6A CapSense コントローラー)	CY3280-20x66	SmartSense 自動チューニング ユーザーモジュール付き CapSense ボタンとスライダーを実装する方法を示す
2	I2C によるホスト通信(CY8C20xx6A CapSense コントローラー使用)	CY3280-20x66	I^2C を介して CapSense コントローラーへ／から CapSense データを転送する方法を示す。また、Bridge Control Panel ツールを使ってデータをプロットし、読み出す方法を説明
3	UART によるデータ送信 (CY8C20xx6A CapSense コントローラー使用)	CY3280-20x66	RS232 データ パケット形式で CapSense データを送信する方法を示す。また、MultiChart ツールを使ってデータをプロットし、読み出す方法を説明
4	絶対センサー静電容量の測定 (CY8C20xx6A CapSense コントローラー使用)	CY3280-20x66	寄生容量 (C_P) が最大 60pF までのセンサーの絶対静電容量を 1pF 以内の精度で測定して、その結果を Windows のハイパーターミナルあるいはブリッジ コントロール パネルを用いて PC に表示
5	絶対センサー静電容量の測定 (CY8C21x34/B CapSense コントローラー使用)	CY3280-21x34	寄生容量 (C_P) が最大 60pF までのセンサーの絶対静電容量を 1pF 以内の精度で測定して、その結果をウィンドウのハイパーターミナルあるいはブリッジ コントロール パネルを用いて PC に表示する
6	絶対センサー静電容量の測定 (CY8C20x34 CapSense コントローラー使用)	CY3280-20x34	寄生容量 (C_P) が最大 60pF までのセンサーの絶対静電容量を 1pF 以内の精度で測定
7	CSD の調整 (CY8C20xx6A CapSense コントローラー使用)	CY3280-20x66	CY8C20xx6A で CSD ユーザーモジュールを調整
8	フィードバック抵抗付き CSD の調整 (CY8C21x34/B CapSense コントローラー使用)	CY3280-21x34	CY8C21x34/B で CSD ユーザーモジュールを調整
9	CSA_EMC の調整 (CY8C20x34 CapSense コントローラー使用)	CY3280-20x34	CY8C21x34 で CSA_EMC ユーザーモジュールを調整
10	CY8C20xx6A の SmartSense による CapSense ボタンの消費電力の最適化	CY3280-20x66	SmartSense ユーザーモジュールを使用して、センサーの消費電流を 50µA 未満にする方法を説明

表 2. 製品ファミリ別サンプルコード

	製品ファミリ			
機能	CY8C20xx6A	CY8C20x34	CY8C21x34/B	CY8C24x94
SmartSense 付きボタン スライダー	サンプルコード 1	N/A[1]	要移植[2]	N/A[1]
I2C とホスト通信 I^2C	サンプルコード 2	要移植[2]	要移植[2]	要移植[2]
UART によるデータの送信	サンプルコード 3	要移植[2]	要移植[2]	要移植[2]
絶対センサー静電容量測定	サンプルコード 4	サンプルコード 5	サンプルコード 6	要移植[2]
CSD の調整	サンプルコード 7	N/A[3]	無	N/A[3]

[1] これらのデバイスは、SmartSense UM に対応していません。
[2] サンプルコードは、「他の CapSense デバイスへのサンプルコードの移行」セクションで説明するように、このガイドで提供されるプロジェクトが付録の各デバイスへ移植された後に作動します。
[3] 調整手法は、このデバイスとユーザーモジュールに固有です。

	製品ファミリ			
機能	**CY8C20xx6A**	**CY8C20x34**	**CY8C21x34/B**	**CY8C24x94**
フィードバック抵抗付き CSD の調整	N/A[3]	N/A[3]	サンプルコード 8	N/A[3]
CSA_EMC の調整	N/A[3]	サンプルコード 9	N/A[3]	N/A[3]
SmartSense で消費電力の最適化	サンプルコード 10	無	無	無

サンプルコード 1　SmartSense 付きボタン スライダー (CY8C20xx6A CapSense コントローラー)

1.1　プロジェクト名

CE_1_ButtonsandSliders_withSmartSense_withCY8C20xx6A

1.2　概要

静電容量センサーを適切に動作させるには、寄生容量 (C_P) に応じていくつかのパラメーターを手動で調整する必要があります。SmartSense™ユーザ モジュールは、CSD UM の基本アーキテクチャを改善、拡張します。主要な機能の 1 つは、SmartSense 自動チューニングです。SmartSense のユーザーモジュールは、各センサーの C_P に基づき、実行中に寄生容量の動作パラメーターを最適化します。SmartSense を使用する際に、手動で調整する必要はありません。このサンプルコードは、SmartSense ユーザーモジュールの使用方法を示します。EzI2C ユーザーモジュールを使用して、CapSense® パラメーター (raw カウント、差分カウント、各ボタンのベースライン、スライダのセントロイド位置など) を I^2C マスターに送信します。

1.3　ハードウェアのセットアップ

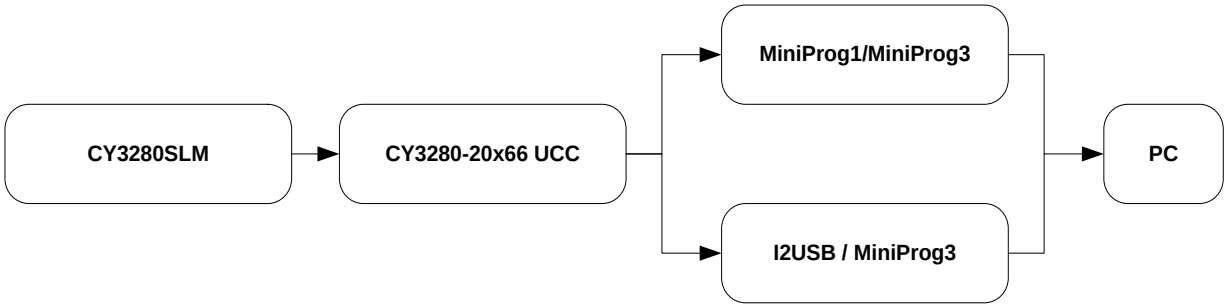
1.3.1　要求事項

- CY3280-20x66 Universal CapSense Controller board
- CY3280-SLM Universal CapSense Linear Slider Module
- CY3240-I2USB Bridge または CY8CKIT-002 Miniprog3
- CY3217-MiniProg1 Programmer Kit または CY8CKIT-002 Miniprog3
- USB Aーミニ B のケーブル
- Windows XP 以降のオペレーティング システムを実行する PC

1.3.2　組み立て

図 1-1 ハードウェアの準備を示します。CY3280-20x66 UCC キットを CY3280-SLM モジュールに 22x2_RA_Receptacle 経由で接続します。MiniProg1/MiniProg3 または I2USB ブリッジ/MiniProg3 を使って、キットの ISSP ヘッダーに接続します。ハードウェア準備では、プログラミングに MiniProg1/MiniProg3 を使用し、PC にデータを送信するには I2USB ブリッジ/Miniprog3 を使用します。これらは、USB ケーブルで PC に接続します。

図 1-1. ハードウェア セットアップのブロック図

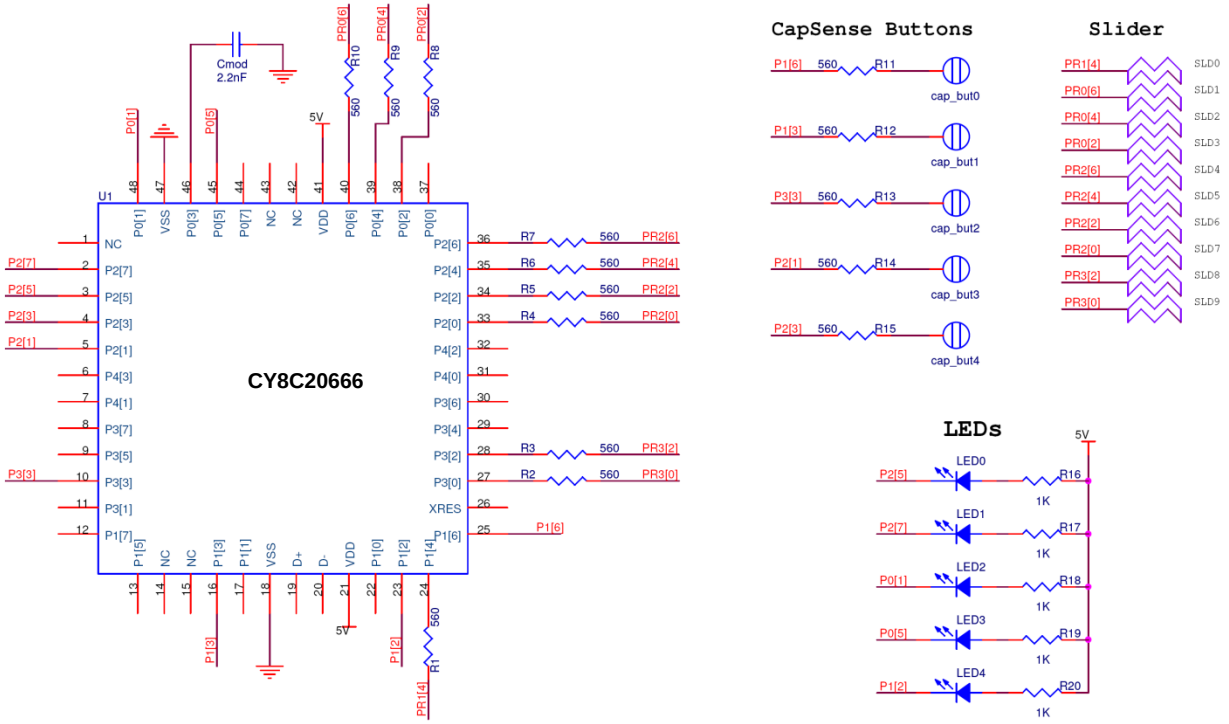


1.3.3 ボードの設定

以下のハードウェアを接続します。

- CY3280-SLM ドーター カードのヘッダーJ1 を CY3280-20x66 UCC ボード上の 22x2_RA_Receptacle (コネクタ P2) に接続します。
- ヘッダーJ7 にジャンパーを嵌め、UCC ボードの V_{CC} ピンと V_{CC_PROG} ピンを接続させます。この設定により、ISSP コネクタから CapSense コントローラーに電力を供給できます。
- ヘッダーJ4 にジャンパーを嵌め、UCC ボードの XRES ピンと XRES/INT ピン (ピン 1 と 2) を接続させます。この設定では、CapSense コントローラーの XRES ピンが ISSP コネクタ J3 のピン 3 に接続されます。
- ヘッダーJ2 にジャンパーを嵌め、CY3280-SLM ボードの GND ピンと SHILED ピン (ピン 2 と 3) を接続させます。この設定により、CY3280-SLM ボード上のハッチ パターンが接地されます。
- MiniProg1/MiniProg3 を UCC ボードの J3 上の ISSP ヘッダーに接続します。この接続は、コードで生成される hex ファイルで UCC をプログラミングする時のみ要求されます。この接続は、CapSense データが Bridge Control Panel ソフトウェアで読み出される時、I2USB ブリッジ/MiniProg3 と置き換える必要があります。
- USB Aーミニ B ケーブルを使用して、MiniProg1 (または I2USB ブリッジまたは MiniProg3) 他方の端を PC に接続します。

1.4 回路図



変調コンデンサ (C_{MOD}) は、P0[3] に接続された 2.2nF のコンデンサです。560Ω の抵抗は、RF 干渉を軽減するため、各 CapSense ボタンに直列に接続されます。LED は 1kΩ の直列抵抗に接続しアクティブローの設定です。全部で、LED が 5 個、ボタンが 5 個、セグメントが 10 あるスライダが 1 個あります。

表 1-1. LED、ボタン、スライダー セグメントのピン割り当て

LED	ボタン	スライダー セグメント
LED0 - P2[5]	BTN0 - P1[6]	SLD0 - P1[4]、SLD1 - P0[6]
LED1 - P2[7]	BTN1 - P1[3]	SLD2 - P0[4]、SLD3 - P0[2]
LED2 - P0[1]	BTN2 - P3[3]	SLD4 - P2[6]、SLD5 - P2[4]
LED3 - P0[5]	BTN3 - P2[1]	SLD6 - P2[2]、SLD7 - P2[0]
LED4 - P1[2]	BTN4 - P2[3]	SLD8 - P3[2]、SLD9 - P3[0]

1.5 ソフトウェアの準備

1.5.1 ツール

- PSoC Designer™ (バージョン 5.2 以降)
- PSoC Programmer (バージョン 3.13 以降)
- Bridge Control Panel (ブリッジ コントロール パネル)

1.5.2 ユーザーモジュール リスト

以下の表は、このサンプルコードで使用されるユーザーモジュール (UM)、および各 UM が占有するハードウェア リソースを一覧にします。

ユーザーモジュール	ハードウェア リソース
SmartSense	CapSense とコンパレータ、タイマー 1
EzI2Cs	I²C/SPI ブロック

1.5.3 ユーザーモジュール パラメーター、グローバル リソース

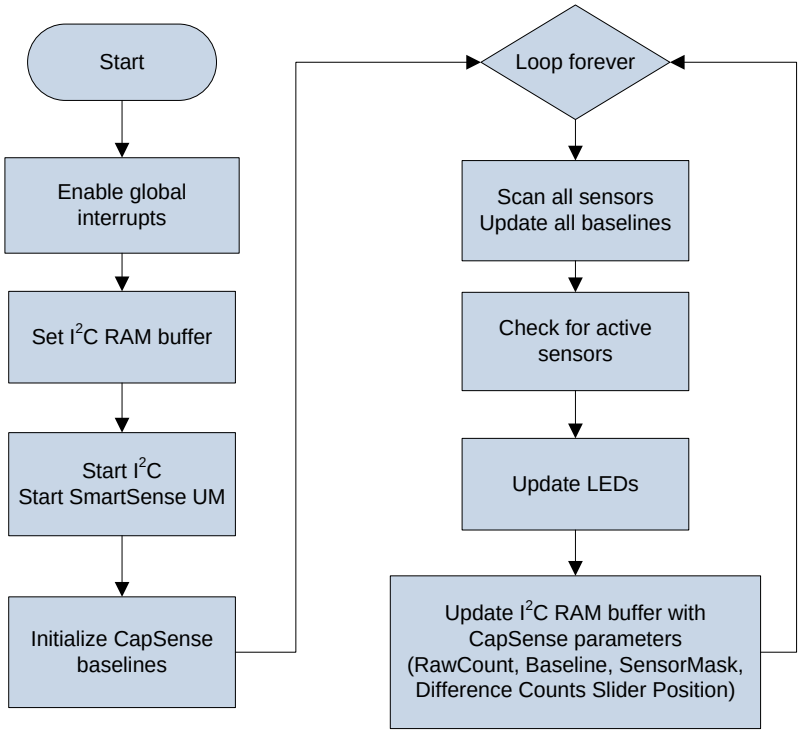
プロジェクトの `ReadMe.txt` ファイルでは、このサンプルコードで使用されるユーザーモジュールのパラメーター設定について説明しています。また、グローバル リソースのリストが含まれています。

1.6 動作

リセットすると、ファームウェアは以下の動作を実行します。

- 構造体 MyI2C_Regs を定義してボタン番号、raw カウント、差分カウント、ベースライン、セントロイド位置、所定のボタン番号に対応する CapSense ボタンの状態を保管します。
- グローバル割り込みを有効にしてから、SmartSense ユーザーモジュールを開始します。
- EzI2Cs UM を開始し、構造体 MyI2C_Regs を I²C RAM バッファとして設定します。
- 無限ループで以下の操作を実行します。
  - すべてのセンサーを継続的にスキャンし、構造体 MyI2C_Regs を raw カウント、差分カウント、ベースライン、スライダのセントロイド位置、および要求される CapSense ボタンで更新します。I²C マスターは、ボタン番号を EzI2C スレーブの I²C バッファの最初のバイトに書き込み、特定のボタンの CapSense を要求することができます。
  - ファームウェアはボタンが押されるのを検出すると、対応する LED を点灯させます。ボタンを放すと、LED を消灯させます。
  - スライダーに触れると、ファームウェアは LED を点灯させて接触位置を示します。

図 1-2. サンプルコード 1 の機能フローチャート



1.7 サンプルコードの実行

ボードにサンプル プロジェクトをプログラムしてから、この手順を使用してサンプルコードを実行します。UCC ボードのプログラミングの方法については、CY3280-20x66 キット ガイドの第 5 章を参照してください。

1. ボードを 5V で給電するには、MiniProg1/MiniProg3 または CY3280-20x66 UCC キットガイドで説明している電源のいずれかを使用します。
2. CY3280-SLM モジュール ボードでリニア スライダーに触れます。CY3280-SLM モジュール ボードの対応する LED が点灯します。
3. ボタンに触れます。CY3280-SLM モジュール ボードの対応する LED が点灯します。同時に複数のボタンが機能します。リニア スライダーとボタンも同時に使用できます。

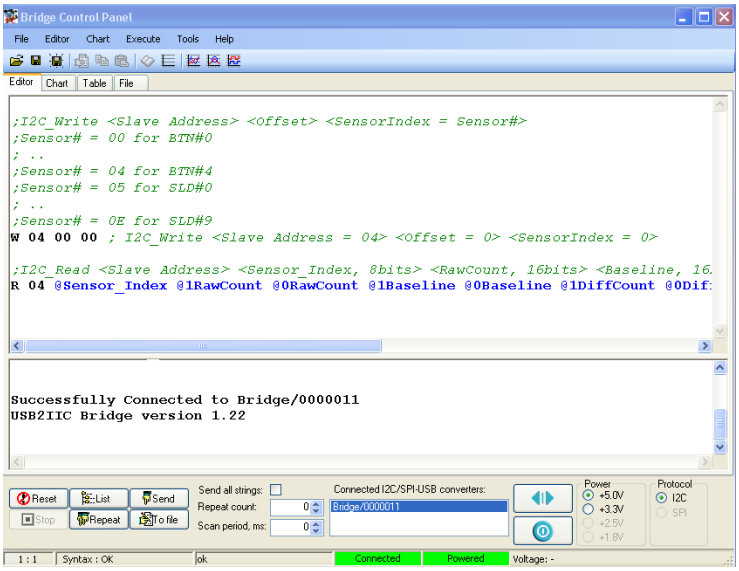
1.8 I²C を経由した CapSense データの読み出し

この手順を使用してサンプルコードを実行し、CapSense データを Bridge Control Panel ツールで読み出します。Bridge Control Panel ツールの詳細については、「AN2397、“CapSense® Data Viewing Tools」を参照してください。

1.8.1 Bridge Control Panel ツールの起動

1. I2USB ブリッジ/MiniProg3 と USB A－ミニ B ケーブルを併用して、コンピューターを CY3280-20x66 ユニバーサル CapSense コントロール ボードの ISSP コネクタ J3 に接続します。
2. コンピューターのデスクトップで、**Start** > **All Programs** > **Cypress** > **Bridge Control Panel** (*バージョン番号*) > **Bridge Control Panel** (*バージョン番号*) を選択します。
   Bridge Control Panel は、PSoC Programmer のインストール中にインストールされるコンポーネントです。
3. ポート選択ウィンドウからデバイスを選択します。
4. CY3280-20X66 CapSense コントローラー ボードに 5V で電力を供給します。
5. Bridge Control Panel で **File** > **Open** を選択します。サンプルコード プロジェクト フォルダに含まれる「BCP Configuration Files」フォルダから `BCP.iic` ファイルを読み出します。

6. **Charts** > **Variable Settings** を選択します。サンプルコード プロジェクト フォルダに含まれる「BCP Configuration Files」フォルダから `BCP.ini` ファイルを読み出します。



7. **OK** をクリックして、メイン画面に戻ります。

1.8.2 BTN0 の raw カウント、ベースライン、差分カウントの読み出し

1. I²C の書き込み命令 W 04 00 00 を一度送信します。
2. **Repeat** ボタンをクリックして、以下の I²C 読み出し命令を継続的に送信します。

```
R 04 @Sensor_Index @1RawCount @0RawCount @1Baseline @0Baseline @1Diffcount @0DiffCount
@ButtonStatus @SliderPosition
```

3. **Chart** タブをクリックして、BTN0 の raw カウント、ベースライン、差分カウントを表示します。

1.8.3 SLD0 の raw カウント、ベースライン、差分カウント、スライダー ポジションの読み出し

1. I²C 書き込み命令 W 04 00 05 を一度送信します。
2. **Repeat** ボタンをクリックして、以下の I²C 読み出し命令を継続的に送信します。

```
R 04 @Sensor_Index @1RawCount @0RawCount @1Baseline @0Baseline @1Diffcount @0DiffCount
@ButtonStatus @SliderPosition
```

3. **Chart** タブをクリックして、SLD0 の raw カウント、ベースライン、差分カウント、スライダー ポジションを表示します。

サンプルコード 2　I²C によるホスト通信 (CY8C20xx6A CapSense コントローラー使用)

2.1　プロジェクト名

CE_2_HostCommunication_viaI2C_withCY8C20xx6A

2.2　概要

このサンプルコードでは、I²C 通信プロトコルを使用して、5 個の CapSense®ボタンと 10 個のセグメントを持つスライダーのデータ (raw カウント、差分カウント、ベースライン、ステータス) を Bridge Control Panel ツールに送信すると、そこでデータが表示されます。CSD アルゴリズムは、CY8C20xx6A CapSense コントローラー上のボタンを実践します。ボタン番号を指定すると、その特定のボタンのデータを取得できます。

2.3　ハードウェアのセットアップ

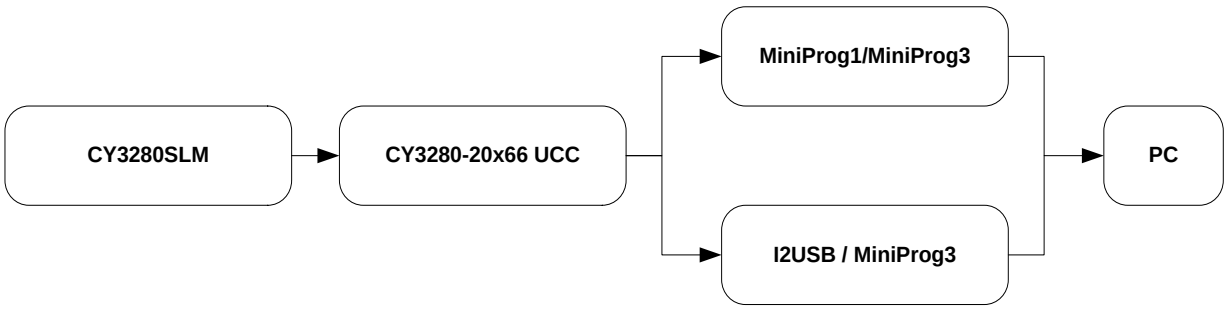
2.3.1　要求事項

- CY3280-20x66 ユニバーサル CapSense コントローラー ボード
- CY3280-SLM ユニバーサル CapSense リニア スライダー モジュール
- CY3240-I2USB ブリッジまたは CY8CKIT-002 Miniprog3
- CY3217-MiniProg1 プログラマ キット または CY8CKIT-002 Miniprog3
- USB Aーミニ B ケーブル

2.3.2　アセンブリ

図 2-1 は、ハードウェアの準備を示します。CY3280-20x66 UCC キットを CY3280-SLM モジュールに 2x2_RA_Receptacle 経由で接続します。MiniProg1/MiniProg3 または I2USB ブリッジ/MiniProg3 をキットの ISSP ヘッダーに接続します。ハードウェア準備では、プログラミングに MiniProg1/MiniProg3 を使用し、PC にデータを送信するには I2USB ブリッジ/MiniProg3 を使用します。これらは、USB ケーブルで PC に接続します。

図 2-1. ハードウェア セットアップのブロック図

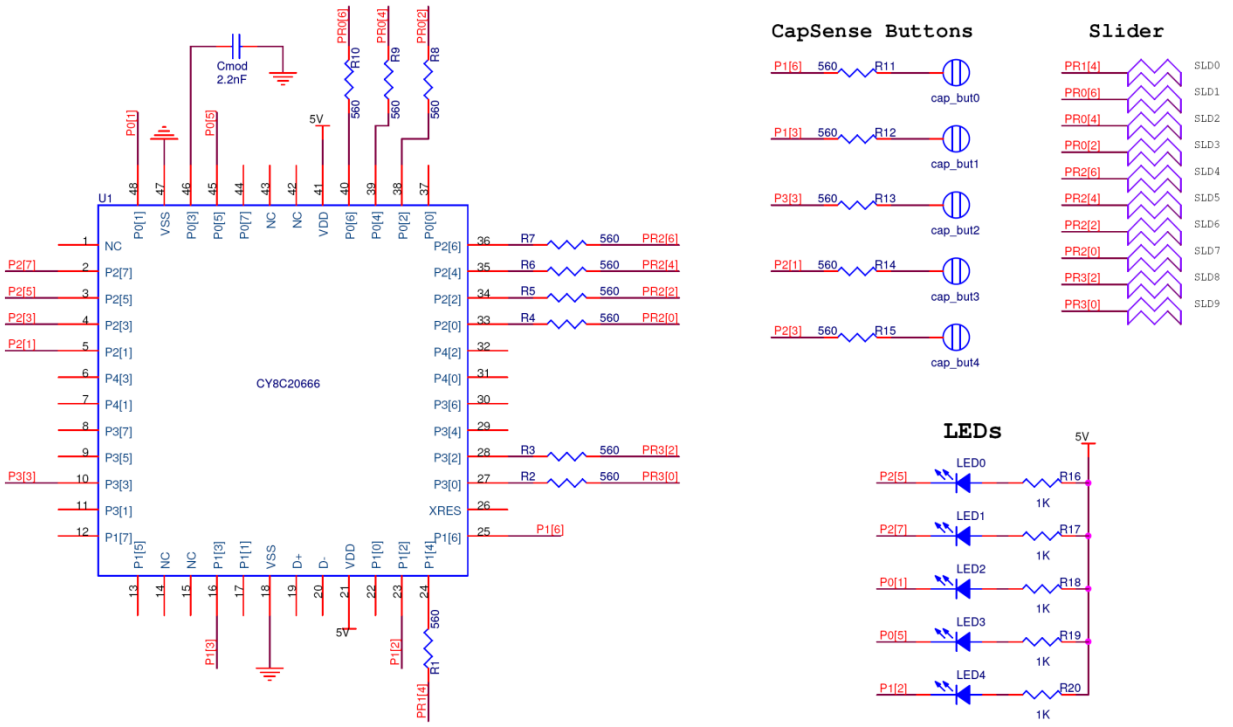


2.3.3 ボードの設定

以下のハードウェアを接続します。

- CY3280-SLM ドーターカードのヘッダーJ1 を CY3280-20x66 UCC ボード上の 22x2_RA_Receptacle (コネクタ P2) に接続します。
- ヘッダーJ7 にジャンパーを嵌め、UCC ボードの V_{CC} ピンと V_{CC_PROG} ピンを接続させます。この設定により、ISSP コネクタから CapSense コントローラーに電力を供給できます。
- ヘッダーJ4 にジャンパーを嵌め、UCC ボードの XRES ピンと XRES/INT ピン (ピン 1 と 2) を接続させます。この設定では、CapSense コントローラーの XRES ピンが ISSP コネクタ J3 のピン 3 に接続されます。
- ヘッダーJ2 にジャンパーを嵌め、CY3280-SLM ボードの GND ピンと SHILED ピン (ピン 2 と 3) を接続させます。この設定により、CY3280-SLM ボード上のハッチパターンが接地されます。
- MiniProg1/MiniProg3 を UCC ボードの J3 上の ISSP ヘッダーに接続します。この接続は、コードで生成される hex ファイルで UCC をプログラミングする時のみ要求されます。この接続は、CapSense データが Bridge Control Panel ソフトウェアで読み出される時、I2USB ブリッジ/MiniProg3 と置き換える必要があります。
- USB A－ミニ B ケーブルを使用して、MiniProg1 (または I2USB ブリッジまたは MiniProg3) 他方の端を PC に接続します。

2.4 回路図



変調コンデンサ (C_{MOD}) は、P0[3] に接続された 2.2nF のコンデンサです。560Ω の抵抗は、RF 干渉を軽減するため、各 CapSense ボタンに直列に接続されます。LED は 1kΩ の直列抵抗に接続しアクティブ ローの設定です。全部で、LED が 5 個、ボタンが 5 個、セグメントが 10 あるスライダーが 1 個あります。

表 2-1. LED、ボタン、スライダー セグメントのピン割り当て

LED	ボタン	スライダー セグメント
LED0 - P2[5]	BTN0 - P1[6]	SLD0 - P1[4]、SLD1 - P0[6]
LED1 - P2[7]	BTN1 - P1[3]	SLD2 - P0[4]、SLD3 - P0[2]

LED2 - P0[1]	BTN2 - P3[3]	SLD4 - P2[6]、SLD5 - P2[4]
LED3 - P0[5]	BTN3 - P2[1]	SLD6 - P2[2]、SLD7 - P2[0]
LED4 - P1[2]	BTN4 - P2[3]	SLD8 - P3[2]、SLD9 - P3[0]

2.5 ソフトウェアの準備

2.5.1 ツール

- PSoC Designer (バージョン 5.2 以降)
- PSoC Programmer (バージョン 3.13 以降)
- Bridge Control Panel (ブリッジ コントロール パネル)

2.5.2 ユーザーモジュール リスト

以下の表は、このサンプルコードで使用されるユーザーモジュール (UM)、および各 UM が占有するハードウェア リソースを一覧にします。

ユーザーモジュール	ハードウェア リソース
CSD	CapSense ブロック、タイマー1 (デフォルト)
EzI2C	I²C/SPI ブロック

2.5.3 ユーザーモジュール パラメーター、グローバル リソース

プロジェクトの `ReadMe.txt` ファイルでは、このサンプルコードで使用されるユーザーモジュールのパラメーター設定について説明しています。また、グローバルリソースのリストが含まれています。

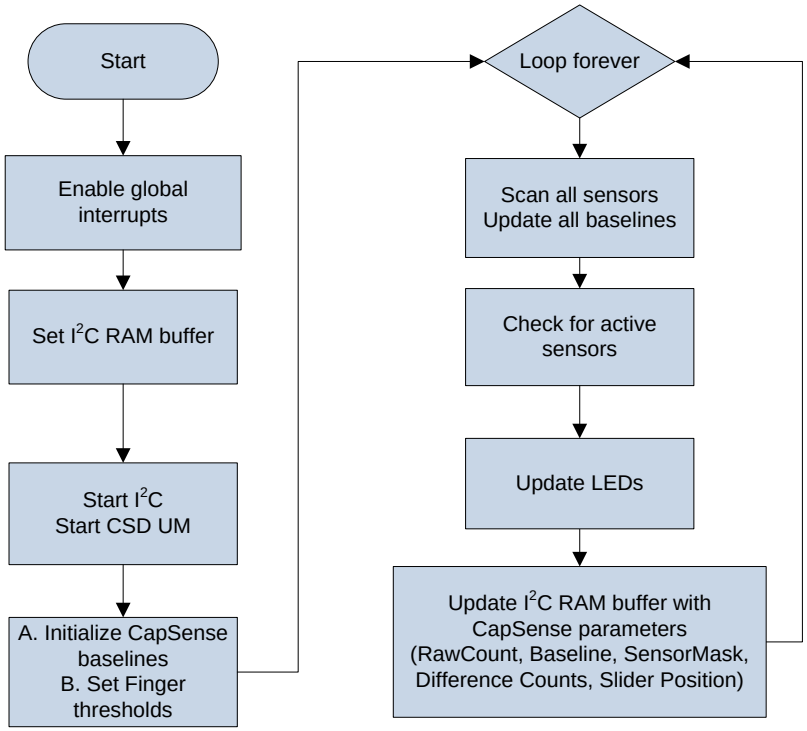
注: サンプルコードは、CY3280-SLM キットに付属するアクリル製オーバーレイ (1.5mm) 向けに調整しました。オーバーレイがこれより厚い場合は、サンプルコード 7: CSD の調整 (CY8C20xx6A CapSense コントローラー使用)で説明しているプロジェクトを調整します。

2.6 動作

リセットすると、ファームウェアは以下の動作を実行します。

- 構造体 MyI2C_Regs を定義してボタン番号、raw カウント、差分カウント、ベースライン、セントロイド位置、所定のボタン番号に対応する CapSense ボタンの状態を保管します。
- グローバル割り込みを有効にしてから、CSD のユーザーモジュールを開始します。
- EzI2C のユーザーモジュールを開始し、構造体 MyI2C_Regs を I²C RAM バッファとして設定します。
- 無限ループで以下の操作を実行します。
  - すべてのセンサーを継続的にスキャンし、構造体 MyI2C_Regs を raw カウント、差分カウント、ベースライン、スライダーのセントロイド位置、および要求される CapSense ボタンで更新します。I²C マスターは、ボタン番号を EzI2C スレーブの I²C バッファの最初のバイトに書き込んで、特定のボタンの CapSense データを要求することができます。
  - ファームウェアはボタンが押されるのを検出すると、対応する LED を点灯させます。ボタンを放すと、LED を消灯させます。
  - スライダーに触れると、ファームウェアは LED を点灯させて接触位置を示します。

図 2-2. サンプルコード 2 の機能フローチャート



2.7 サンプルコードの実行

ボードにサンプル プロジェクトをプログラムしてから、この手順を使用してサンプルコードを実行します。UCC ボードのプログラミングの方法については、CY3280-20x66 キット ガイドの第 5 章を参照してください。

1. ボードを 5V で給電するには、MiniProg1/Miniprog3 または CY3280-20x66 UCC キットガイドで説明している電源のいずれかを使用します。
2. CY3280-SLM モジュール ボードでリニア スライダーに触れます。
   CY3280-SLM モジュール ボードの対応する LED が点灯します。
3. ボタンに触れます。
   CY3280-SLM モジュール ボードの対応する LED が点灯します。同時に複数のボタンが機能します。リニア スライダーとボタンも同時に使用できます。

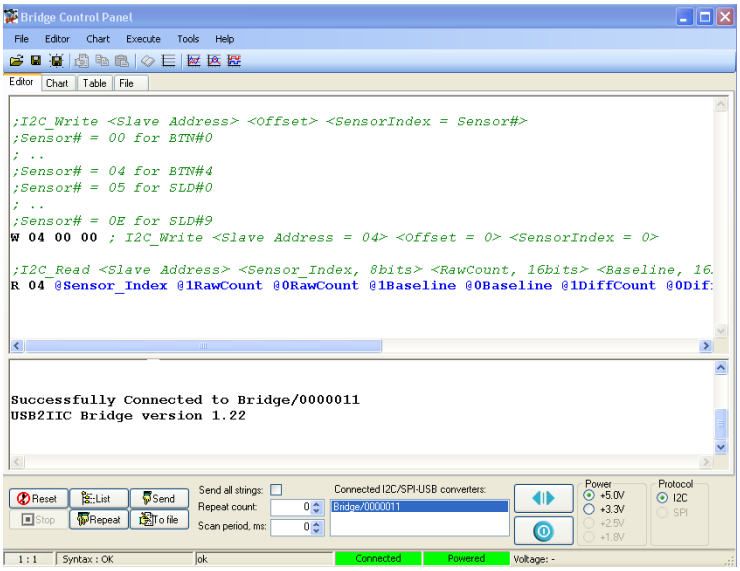
2.8 I²C を経由した CapSense データの読み出し

この手順を使用してサンプルコードを実行し、CapSense データを Bridge Control Panel ツールで読み出します。Bridge Control Panel ツールの詳細については、「AN2397、“CapSense® Data Viewing Tools」を参照してください。

2.8.1 Bridge Control Panel の起動

1. I2USB ブリッジ/MiniProg3 と USB A－ミニ B ケーブルを併用して、コンピューターを CY3280-20x66 ユニバーサル CapSense コントロール ボードの ISSP コネクタ J3 に接続します。
2. コンピューターのデスクトップで、**Start** > **All Programs** > **Cypress** > **Bridge Control Panel** *(バージョン番号)* > **Bridge Control Panel** *(バージョン番号)* を選択します。
   Bridge Control Panel は、PSoC Programmer のインストール中にインストールされるコンポーネントです。
3. ポート選択ウィンドウからデバイスを選択します。
4. CY3280-20X66 CapSense コントローラー ボードに 5V で電力を供給します。
5. Bridge Control Panel で **File** > **Open** を選択します。サンプルコード プロジェクトフォルダに含まれる「BCP Configuration Files」フォルダから `BCP.iic` ファイルを読み出します。

6. **Charts** > **Variable Settings** を選択します。サンプルコード プロジェクト フォルダに含まれる「BCP Configuration Files」フォルダから `BCP.iic` ファイルを読み出します。



7. **OK** をクリックして、メイン画面に戻ります。

2.8.2 BTN0 の raw カウント、ベースライン、差分カウントの読み出し

1. I²C の書き込み命令 W 04 00 00 を一度送信します。
2. **Repeat** ボタンをクリックして、以下の I²C 読み出し命令を継続的に送信します。

```
R 04 @Sensor_Index @1RawCount @0RawCount @1Baseline @0Baseline @1DiffCount @0DiffCount
@ButtonStatus @SliderPosition
```

3. **Chart** タブをクリックして、BTN0 の raw カウント、ベースライン、差分カウントを表示します。

2.8.3 SLD0 の raw カウント、ベースライン、差分カウント、スライダー ポジションの読み出し

1. I²C 書き込み命令 W 04 00 05 を一度送信します。
2. **Repeat** ボタンをクリックして、以下の I²C 読み出し命令を継続的に送信します。

```
R 04 @Sensor_Index @1RawCount @0RawCount @1Baseline @0Baseline @1DiffCount @0DiffCount
@ButtonStatus @SliderPosition
```

3. **Chart** タブをクリックして、SLD0 の raw カウント、ベースライン、差分カウント、スライダー ポジションを表示します。

サンプルコード 3　UART によるデータ送信 (CY8C20xx6A CapSense コントローラー使用)

3.1　プロジェクト名

CE_3_DataTransmission_viaUART_withCY8C20xx6A

3.2　概要

このサンプルコードでは、UART 通信プロトコルを使用して、CapSense®データを MultiChart ツールが読み出せる形式で PC に送信します。MultiChart ツールは、raw カウント、差分カウント (信号)、ベースライン、およびスライダの位置の CapSense パラメーターをグラフ形式で表示するのに使用します。SmartSense™ ユーザーモジュール (UM) は、CY3280-SLM ですべてのボタンとスライダー セグメントを継続的にスキャンし、UART UM は CapSense パラメーターを TX8 出力ピンに送信します。外部レベル トランスレータは、TX8 データを RS232 レベルに変換してから、標準 RS232 ケーブルで PC に送信します。PC では、MultiChart ツール (AN2397 で紹介) を使用してデータを監視します。このサンプルコードは、CY3280-SLM の LED のオンとオフを切り替えて、ボタンのステータスやスライダーの位置を示します。

3.3　ハードウェアの準備

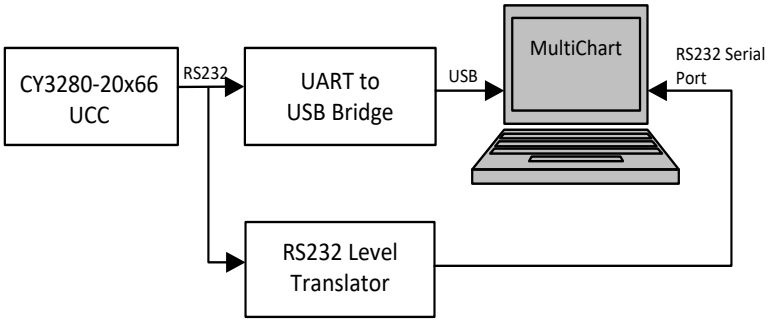
3.3.1　要求事項

- CY3280-20x66 ユニバーサル CapSense コントローラー ボード
- CY3280-SLM ユニバーサル CapSense リニア スライダー モジュール
- CY3217-MiniProg1 プログラマ キット または CY8CKIT-002 Miniprog3
- USB A－ミニ B ケーブル
- 外部 RS232 レベル トランスレータ
- 標準 9 ピン RS232 のシリアル ケーブル
- シリアル ポート (9 ピン RS232) 搭載 PC

3.3.2　アセンブリ

図 3-1 ハードウェアの準備を示します。CY3280-20x66 UCC キットを CY3280-SLM モジュールに 22x2_RA_Receptacle 経由で接続します。MiniProg1/MiniProg3 をヘッダーJ3 に接続してから、キットをプログラミングし、電力を供給します。コネクタ P3 の P3[5]は、外部レベル トランスレータに接続してから UART ケーブル経由で PC に接続します。

図 3-1. ハードウェア セットアップのブロック図



注: CY3280-20x66 UCC は、UART －USB ブリッジまたは RS232 レベル トランスレータ (RS232 シリアル ポート搭載の PC が必要) のいずれかの方法で PC にデータを送信します。

3.3.3 ボードの設定

以下のハードウェアを接続します。

- CY3280-SLM ドーターカードのヘッダーJ1 を CY3280-20x66 UCC ボード上の 22x2_RA_Receptacle (コネクタ P2) に接続します。
- ヘッダーJ7 にジャンパーを嵌め、UCC ボードの V_{CC} ピンと V_{CC_PROG} ピンを接続させます。この設定により、ISSP コネクタから CapSense コントローラーに電力を供給できます。
- ヘッダーJ4 にジャンパーを嵌め、UCC ボードの XRES ピンと XRES/INT ピン (ピン 1 と 2) を接続させます。この設定では、CapSense コントローラーの XRES ピンが ISSP コネクタ J3 のピン 3 に接続されます。
- ヘッダーJ2 にジャンパーを嵌め、CY3280-SLM ボードの GND ピンと SHILED ピン (ピン 2 と 3) を接続させます。この設定により、CY3280-SLM ボード上のハッチパターンが接地されます。
- MiniProg1/ MiniProg3 を J3 ヘッダー上の CY3280-20x66 UCC に接続します。
- CY3280-20x66 UCC 上のコネクタ P3 のコネクタ P3[5]を外部レベル トランスレータに接続します。サンプルの回路図については、図 3-2 を参照してください。
- CY3280-20x66 UCC 上のコネクタ P3 から GND を外部レベル トランスレータに接続します。
- 外部レベル トランスレータに電力を供給します。CY3280-20x66 上のコネクタ P3 から V_{CC} を使用して、外部レベルの返還装置に 5V の電力を供給します。
- RS232 ケーブルを外部レベル トランスレータの出力から PC に接続させます。

3.4 回路図

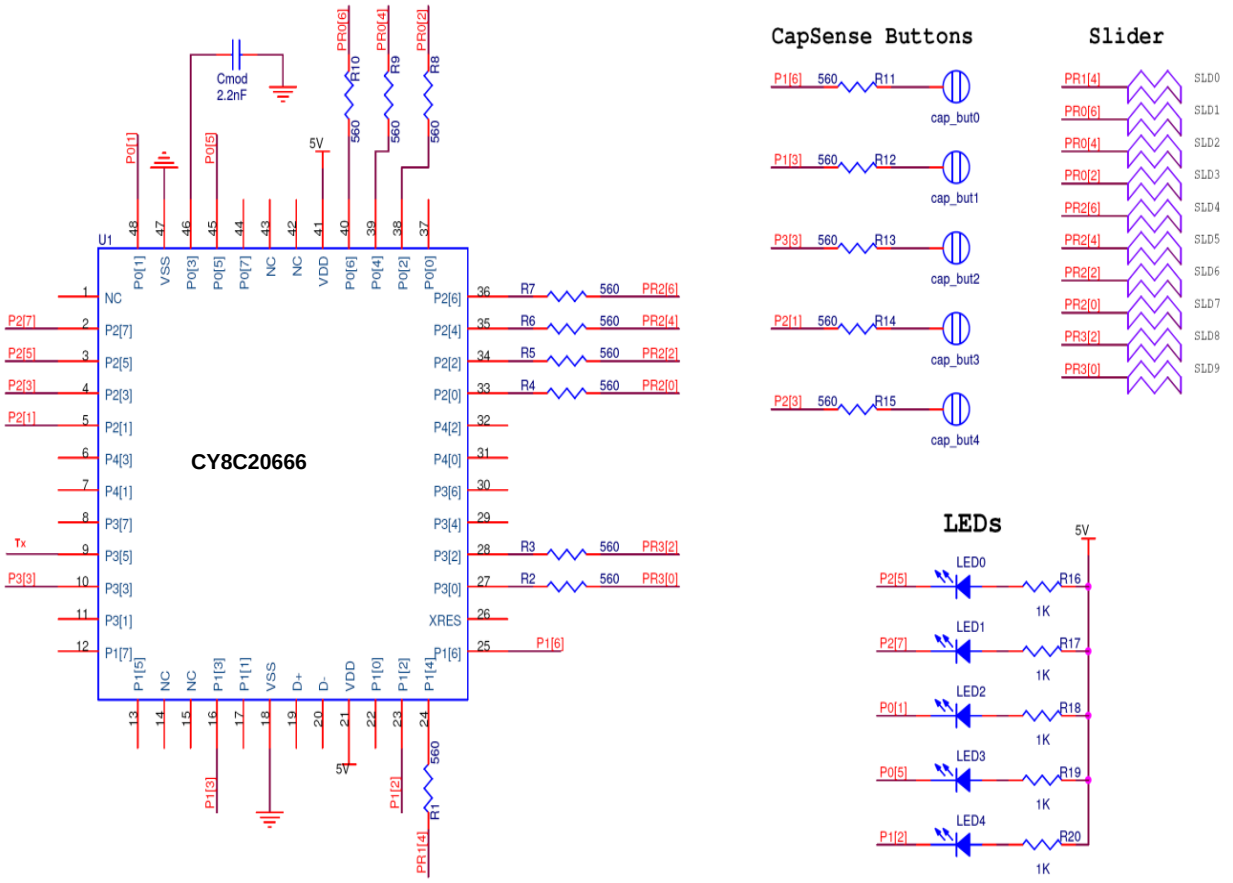
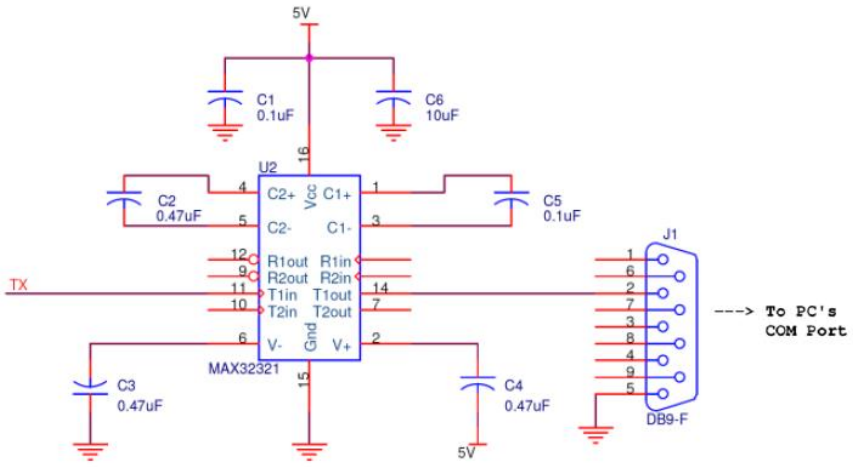


図 3-2. 外部レベル トランスレータ



変調コンデンサ (C_{MOD}) は、P0[3] に接続された 2.2nF のコンデンサです。560Ω の抵抗は、RF 干渉を軽減するため、各 CapSense ボタンに直列に接続されます。LED は 1kΩ の直列抵抗に接続しアクティブローの設定です。全部で、LED が 5 個、ボタンが 5 個、セグメントが 10 あるスライダーが 1 個あります。

表 3-1. LED、ボタン、スライダー セグメントのピン割り当て

LED	ボタン	スライダー セグメント
LED0 - P2[5]	BTN0 - P1[6]	SLD0 - P1[4]、SLD1 - P0[6]
LED1 - P2[7]	BTN1 - P1[3]	SLD2 - P0[4]、SLD3 - P0[2]
LED2 - P0[1]	BTN2 - P3[3]	SLD4 - P2[6]、SLD5 - P2[4]
LED3 - P0[5]	BTN3 - P2[1]	SLD6 - P2[2]、SLD7 - P2[0]
LED4 - P1[2]	BTN4 - P2[3]	SLD8 - P3[2]、SLD9 - P3[0]

3.5 ソフトウェアの準備

3.5.1 ツール

- PSoC Designer (バージョン 5.2 以降)
- PSoC Programmer (バージョン 3.13 以降)
- MultiChart ツール (AN2397 で紹介)

3.5.2 ユーザーモジュールの一覧

以下の表は、このサンプルコードで使用されるユーザーモジュール (UM)、および各 UM が占有するハードウェア リソースを列挙しています。

ユーザーモジュール	ハードウェア リソース
SmartSense	CapSense ブロック、タイマー1 (デフォルト)
UART	占有するブロックなし (ソフトウェア実装)

3.5.3 ユーザーモジュール パラメーター、グローバル リソース

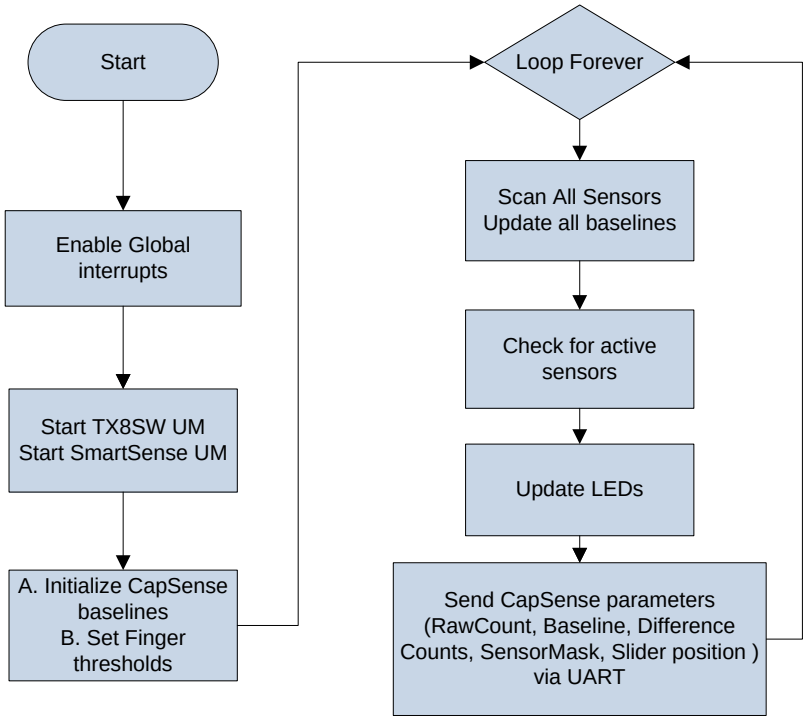
プロジェクトの `ReadMe.txt` ファイルでは、このサンプルコードで使用されるユーザーモジュールのパラメーター設定について説明しています。また、グローバル リソースのリストが含まれています。

3.6 動作

リセットすると、プログラムはデバイス コンフィギュレーションからデバイスにすべてのハードウェア設定を読み出し、*main.c* を実行します。ファームウェアは以下の動作を実行します。

- グローバル割り込みを有効にしてから、SmartSense と UART ユーザーモジュールを開始します。
- センサー ベースラインを初期化します。
- 無限ループで以下の操作を実行します。
  - すべてのセンサーを継続的にスキャンし、UART ユーザーモジュールを使用して raw カウント、差分カウント、ベースライン、およびスライダーの位置を TX8 ピンに送信します。
  - ファームウェアはボタンが押されるのを検出すると、対応する LED を点灯させます。ボタンを放すと、LED を消灯させます。
  - スライダーに触れると、ファームウェアは LED を点灯させて接触位置を示します。

図 3-3. サンプルコード 3 の機能フローチャート



3.7 コード サンプルの実行

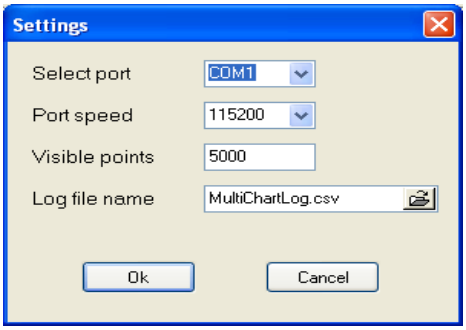
ボードにサンプルプロジェクトをプログラムしてから、この手順を使用してサンプルコードを実行します。UCC ボードのプログラミングの方法については、CY3280-20x66 キット ガイドの第 5 章を参照してください。

1. ボードを 5V で給電するには、MiniProg1/ MiniProg3 または CY3280-20x66 UCC キットガイドで説明している電源のいずれかを使用します。
2. CY3280-SLM モジュール ボードでリニア スライダーに触れます。
   CY3280-SLM モジュール ボードの対応する LED が点灯します。
3. ボタンに触れます。
   CY3280-SLM モジュール ボードの対応する LED が点灯します。同時に複数のボタンが機能します。リニア スライダーとボタンも同時に使用できます。

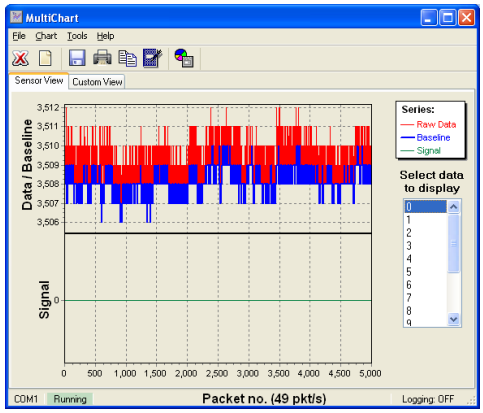
3.8 CapSense データを MultiChart でプロットする

この手順を使用してサンプルコードを実行し、MultiChart ツールをプロットしたデータを表示します。MultiChart ツールの詳細について、およびこのツールをダウンロードするには、「AN2397, “CapSense® Data Viewing Tools”」を参照してください。

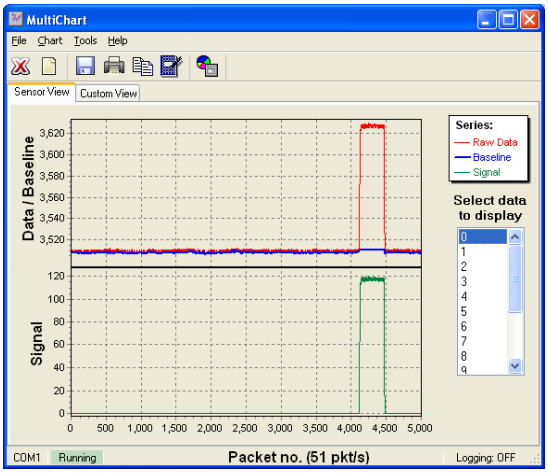
1. MultiChart ツールを開きます。
2. MultiChart ツールで、**Tools** > **Settings** を選択します。
   設定ダイアログ ボックスが表示されます。



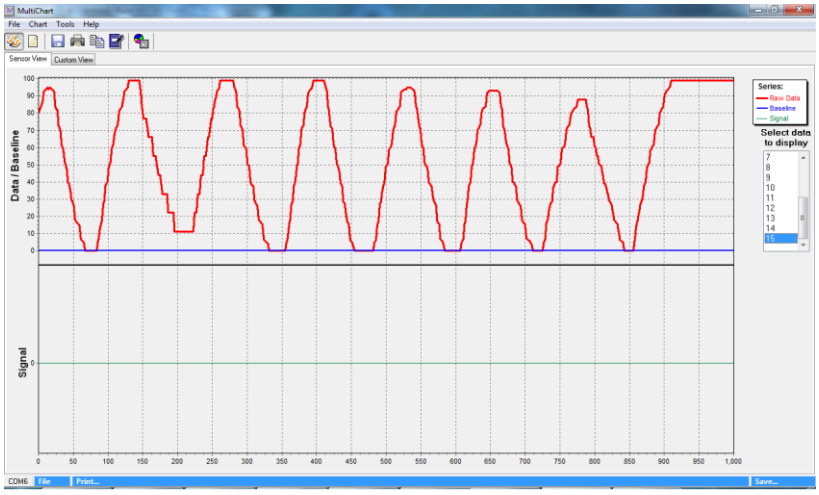
3. **Select port** ドロップリストを使用して、ボードをコンピュータに接続するのに使用する COM ポートを選択します。
   ほとんどの場合、COM1 経由で接続します。ただし、御使用のシステムはこれとは異なるかもしれません。
4. **Port speed** ドロップリストで 115200 を選択してから、**OK** をクリックします。
   raw カウントが赤で、ベースラインがブルーで表示されます。



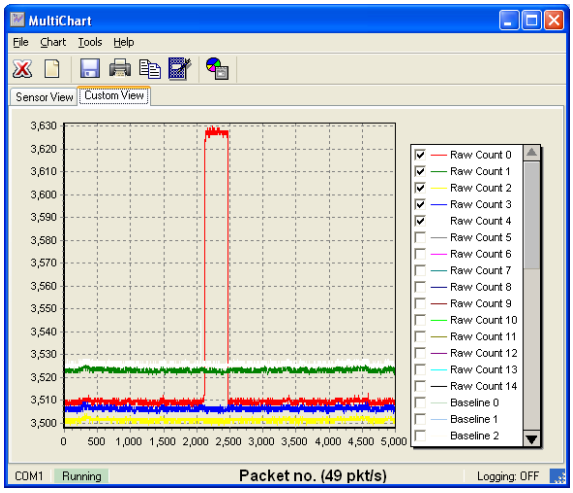
5. ボタン 0 に触れて、信号を観察します (差分カウント)。



6. シリアル 15 の「raw count 15」変数は下記のようにスライダーの位置 (0～100) を示します。観察できない場合、拡大したり、Chart > Edit...>Axis で軸を自動に設定したりします。



7. センサービューでは、一度に 1 台のみセンサーが表示されます。この場合、「Raw Count 15」はスライダーの位置を示します。複数のセンサーを観察するには、「Custom View」タブをクリックします。



サンプルコード 4　絶対センサー静電容量の測定 (CY8C20xx6A CapSense コントローラー使用)

4.1　プロジェクト名

CE_4_MeasuringAbsSensorCap_withCY8C20xx6A

4.2　概要

このサンプルコードでは、CY8C20xx6A CapSense®コントローラーを使用して、5 個のセンサーの絶対静電容量を計算し、Windows のハイパー ターミナル ソフトウェアまたはブリッジ コントロール パネルを搭載したコンピューターで結果を表示する方法を示します。絶対静電容量は 1pF 単位で測定されます。

4.3　ハードウェアの準備

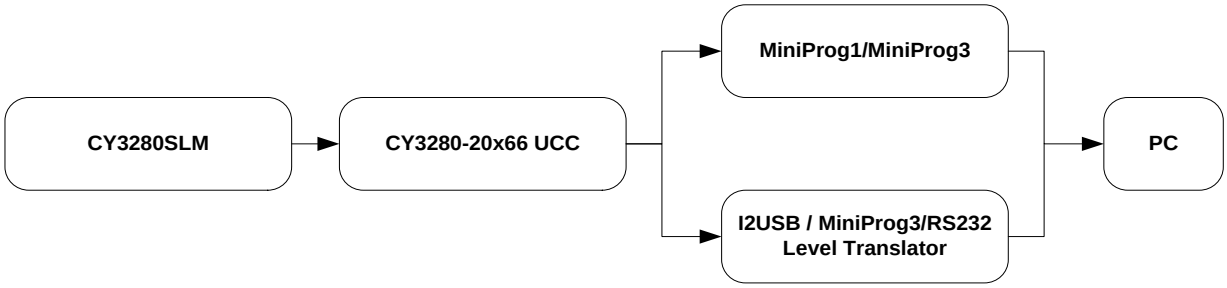
4.3.1　要求事項

- CY3280-20x66 ユニバーサル CapSense コントローラー
- CY3280-SLM ユニバーサル CapSense リニア スライダー モジュール
- RS232 レベル トランスレータ モジュール
- CY3240-I2USB ブリッジまたは CY8CKIT-002 Miniprog3
- CY3217-MiniProg1 プログラマ キット または CY8CKIT-002 Miniprog3
- USB Aーミニ B のケーブル
- Windows XP 以降のオペレーティング システムを実行する PC

4.3.2　アセンブリ

下図はハードウェアの準備を示します。CY3280-20x66 UCC キットを CY3280-SLM モジュールに 22x2_RA_Receptacle 経由で接続します。MiniProg1/MiniProg3 をヘッダーJ3 に接続してから、キットをプログラミングし、電力を供給します。

図 4-1. ハードウェア セットアップのブロック図



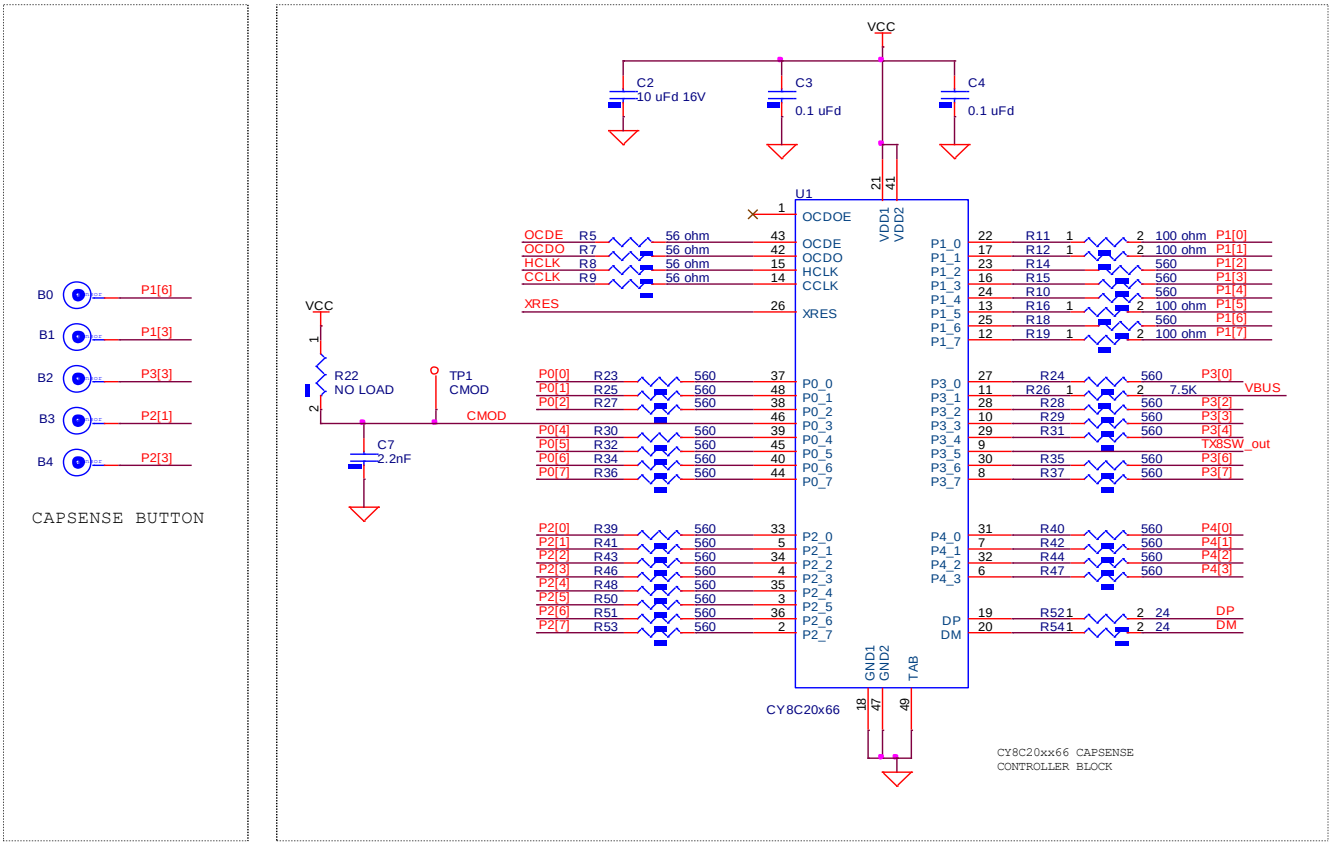
注: このサンプルコードは、外部レベル トランスレータを使用します。ただし、UART−USB ブリッジを使用して、PC でデータを表示することも可能です。UCC キットに含まれる CY3240-I2USB ブリッジを UART−USB ブリッジに変換することができます。詳細については、「AN2397, “CapSense® Data Viewing Tools”」を参照してください。ブリッジ コントロール パネル ソフトウェアは、I2C-USB ブリッジまたは MiniProg3 を介して PC のセンサーの Cp を表示するために使用されます。ハイパーターミナルは RS232 通信プロトコルで PC のデータを表示するために使用されます。

4.3.3 ボードの設定

以下のハードウェアを接続します。

- CY3280-SLM ドーターカードのヘッダーJ1 を CY3280-20x66 UCC ボード上の 22x2_RA_Receptacle (コネクタ P2) に接続します。
- ヘッダーJ7 にジャンパーを嵌め、UCC ボードの V_{CC} ピンと V_{CC_PROG} ピンを接続させます。この設定により、ISSP コネクタから CapSense コントローラーに電力を供給できます。
- ヘッダーJ4 にジャンパーを嵌め、UCC ボードの XRES ピンと XRES/INT ピン(ピン 1 と 2) を接続させます。この設定では、CapSense コントローラーの XRES ピンが ISSP コネクタ J3 のピン 3 に接続されます。
- ヘッダーJ2 にジャンパーを嵌め、CY3280-SLM ボードの GND ピンと SHILED ピン (ピン 2 と 3) を接続させます。この設定により、CY3280-SLM ボード上のハッチパターンがグランド接続されます。
- MiniProg1/MiniProg3 を J3 ヘッダー上の CY3280-20x66 UCC に接続します。
- センサーの Cp が I2C 通信プロトコルによってブリッジ コントロール パネルで表示される場合、CY3280-20x66A のジャンパ J3 を I2C-USB ブリッジ/MiniProg3 に、および USB A ～ ミニ B ケーブルによってブリッジ/MiniProg3 の残りの端をコンピュータに接続させます。ブリッジ コントロール パネルの電源切り替えボタンをクリックしてデバイスに電源を供給します。
- センサーCp が UART によりハイパーターミナルで表示される場合、RS232 レベル トランスレータ ボード上の TX ピンを UCC の P1[7]に接続させます。レベル トランスレータの Vcc ピンと GND ピンを、UCC の Vcc ピンと GND ピンにそれぞれ接続します。RS232 レベル トランスレータの好例は、MAX232 シリアル レベル コンバータです。
- シリアル ケーブルを使用して、レベル トランスレータ ボードの RS232 ポートをコンピューターに接続します。

4.4 回路図



変調コンデンサ (C_{MOD}) は、P0[3] に接続された 2.2nF のコンデンサです。560Ω の抵抗は、RF 干渉を軽減するため、各 CapSense ボタンに直列に接続されます。

表 4-1. CapSense ボタンのピンの割り当て

ボタン	ピン
0	P1[6]
1	P1[3]
2	P3[3]
3	P2[1]
4	P2[3]

4.5 ソフトウェアの準備

4.5.1 ツール

- PSoC® Designer™ (バージョン 5.2 以降)
- PSoC Programmer (バージョン 3.13 以降)
- Bridge Control Panel (ブリッジ コントロール パネル)
- Windows ハイパーターミナル

4.5.2 ユーザーモジュール

以下の表は、このサンプルコードで使用されるユーザーモジュール(UM)、および各ユーザーモジュールが占有するハードウェア リソースを一覧にします。

ユーザー モジュール	ハードウェア リソース
CSD	CapSense ブロック、タイマー1 (デフォルト)
EzI2C	I2C ブロック
TX8SW	占有するブロックなし (ソフトウェア実装)

4.5.3 ユーザーモジュール パラメーター、グローバル リソース

プロジェクトの `ReadMe.txt` ファイルでは、このサンプルコードで使用されるユーザーモジュールのパラメーター設定について説明しています。また、グローバル リソースのリストが含まれています。

4.6 動作

このプログラムでは、センサーの raw カウントの計算に以下の公式を使用します。

$$raw_count = \left(\frac{(2^n - 1) \cdot V_{ref} \cdot \overline{f_s}}{i_{DAC}} \right) \cdot C_s$$

ここでは、

n = 分解能

Vref = リファレンス電圧

fs = プリチャージ クロックのスイッチング電圧

Cs = センサー静電容量

iDAC = iDAC 電流

n、f および Vref は CSD UM パラメーターで確定できます。

iDAC は以下の式で確定できます。

$$i_{DAC} = IDAC_D \cdot iDAC_{gain} \cdot iDAC_{range}$$

IDAC_D は制御レジスタの設定であり、iDACrange は CSD UM パラメーター設定での「Idac 範囲」です。これら両方の値は設計時に既知の値です。IDAC を計算するために必要な唯一の欠けている部分は iDACgain です。iDACgain はデバイスによって PVT と共に変化する A/ビットの単位での iDAC 強度です。4X 範囲では、iDACgain のテスト限界は 230nA/ビット < iDACgain < 270nA/ビットです。

クリプトン ATE プログラム (量産中) は、IDAC_D = 0xC0 = 192 で 4x 範囲内で iDAC 電流を測定します。公称 247nA/ビット iDAC の場合、この電流は 247nA/ビットになります；192 * 4 = 190uA。ATE は実際の電流を測定し、190uA との差を uA の単位および符号と絶対値のフォーマットで記録し、それを bIDACComp として格納します。最上位ビット (MSB) (マスク = 0x80) は符号を表すビットとして解釈され、その値が「1」の場合、 負の値を表し、「0」の場合、「正の値」を表します。例えば、0x8D が格納されたデバイスの場合、iDACgain = (190uA – 13uA) / (4 * 192) = 230nA/ビットとなります。

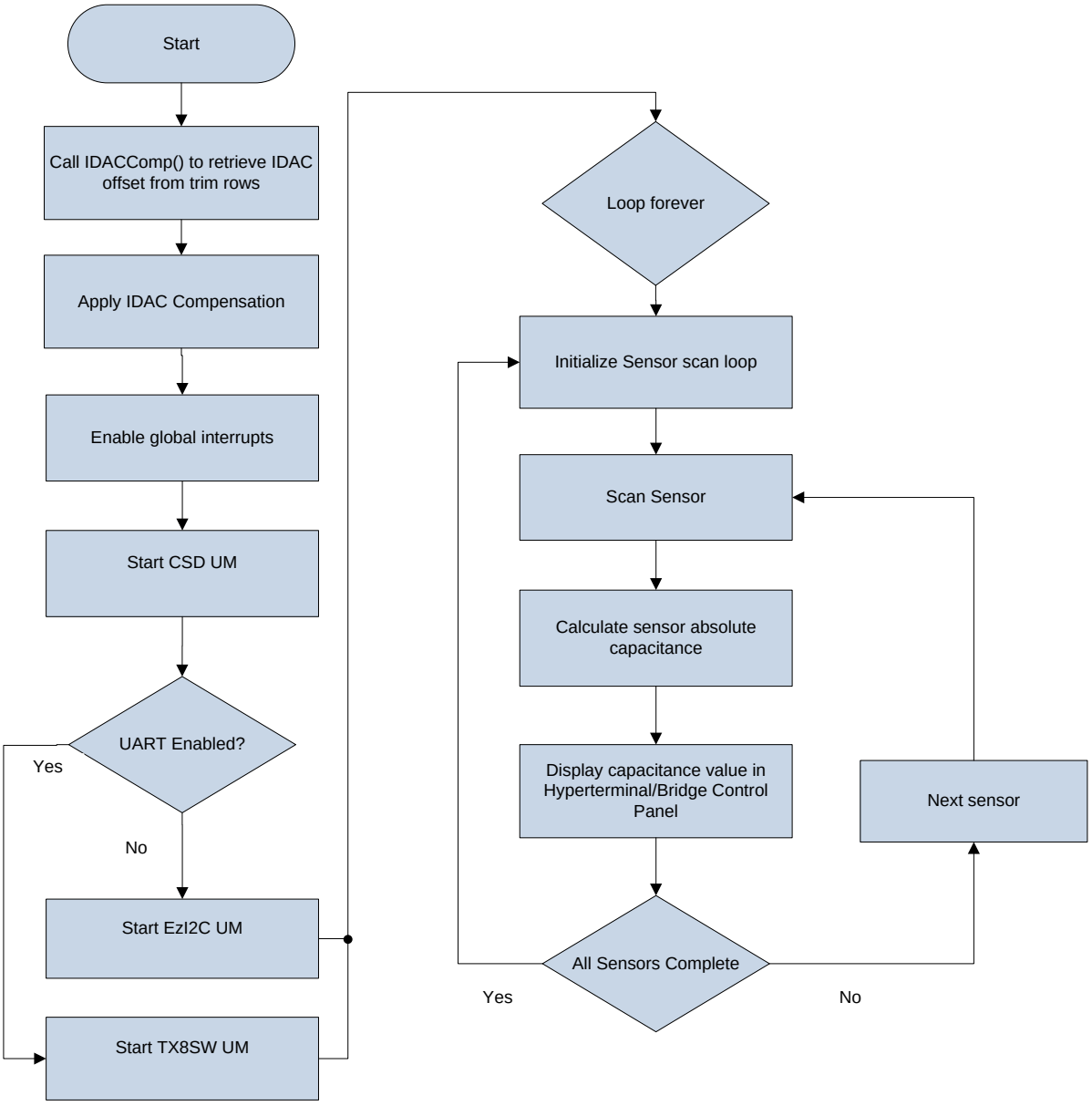
注:

感度= Cs/Raw カウント = $\frac{(2^n - 1) * V_{ref} * f_s}{i_{DAC}}$

このプログラムでは、以下の手順を使用して画面に絶対静電容量を表示します。

1. センサーの感度を計算し、可変感度で結果を保管します。GetIDACComp() は、*GetIDACComp.asm* ファイルで定義されたカスタム機能で、IDAC で部分ごとのばらつきの IDAC 補正を取得します。
2. 注: IDAC は温度のばらつきにより変動するため、このサンプルコードは室温でのみ有効です。
3. グローバル割り込みを有効にしてから、ユーザーモジュールを開始します。
4. 各センサーを順番にスキャンし、配列 CSD_waSnsResult[] から raw カウントを取得します。
5. raw カウントと感度の比率を取得して、絶対静電容量を計算します。
6. ユーザーの選択に応じてブリッジ コントロール パネルまたはハイパーターミナルを表示するために結果をコンピュータに送信します。

図 4-2. サンプルコード 4 のフローチャット



4.7 サンプルコードの実行

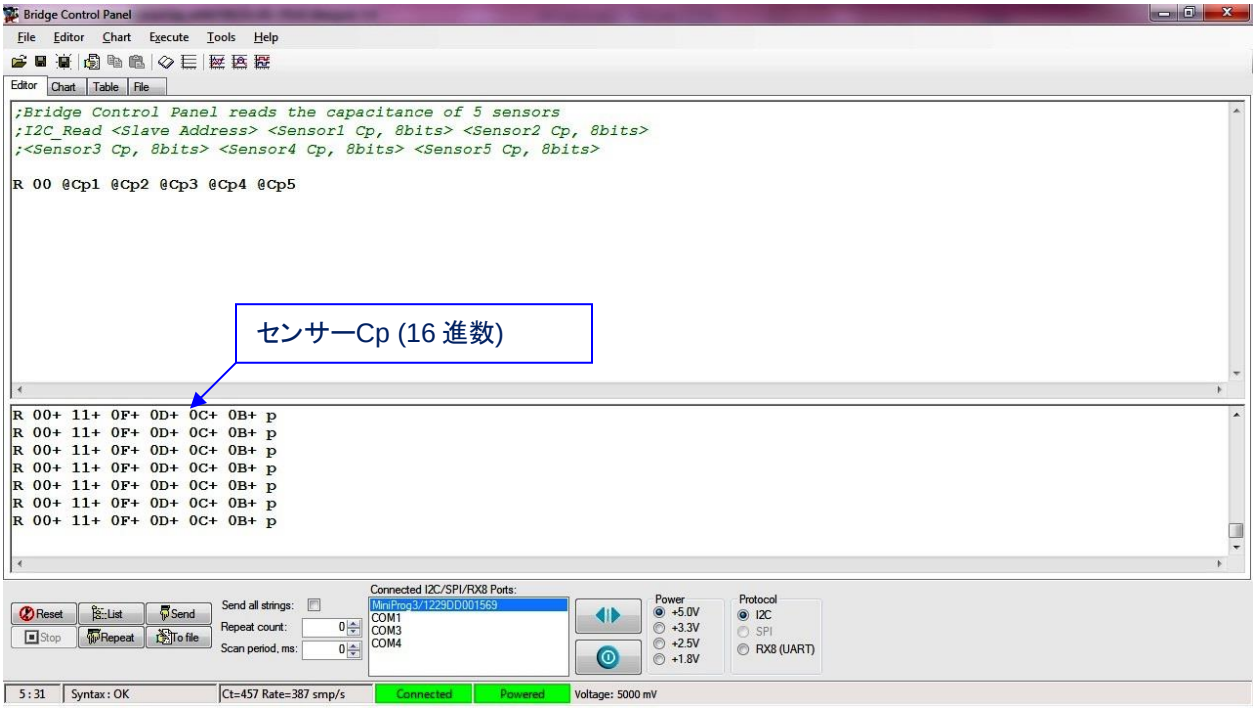
UCC 基板のプログラミング

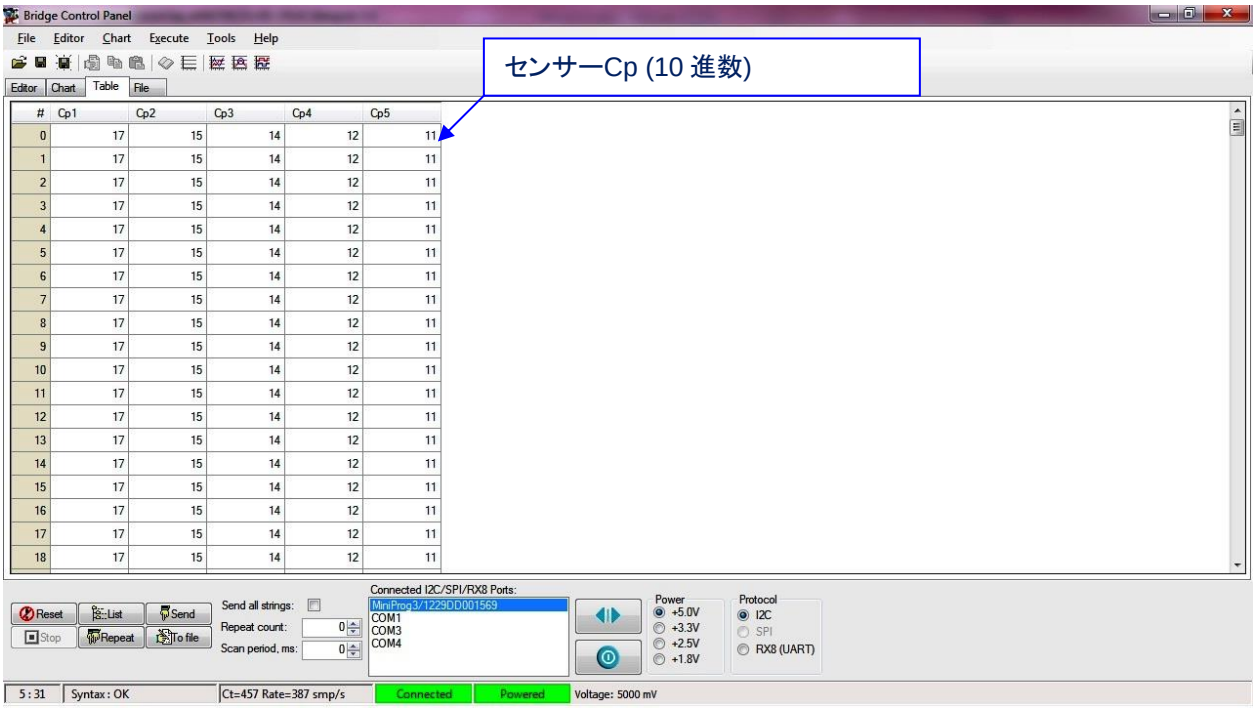
#define UART ステートメントが以下のスクリーンショットに示されるように *main.c* でコメントアウトされます。その後、プロジェクトを生成し (F6 を押す)、ボードをプログラムします。UCC ボードのプログラミングの方法については、CY3280-20x66 キット ガイドの第 5 章を参照してください。

```
74
75  /* Comment this while using I2C;Uncomment this while using UART */
76    //#define UART
77
```

BCP によるセンサー データの観察

1. CY3280-20x66A のジャンパ J3 を I2C-USB ブリッジ/MiniProg3 に、および USB A～ミニ B ケーブルによって I2C-USB ブリッジ/MiniProg3 の残りの端をコンピューターに接続させます。
2. コンピューターのデスクトップで、**Start** > **All Programs** > **Cypress** > **Bridge Control Panel** (バージョン番号) > **Bridge Control Panel** (バージョン番号) を選択します。
3. ポート選択ウィンドウからデバイス (I2C-USB ブリッジまたは MiniProg3) を選択します。
4. ブリッジ コントロール パネルのトグルパワー ボタンをクリックしてデバイスに電源を供給します。
5. Bridge Control Panel ツールで **File** > **Open** を選択します。サンプルコード プロジェクト フォルダに含まれる「BCP Configuration Files」フォルダから *BCP.ini* ファイルを読み出します。
6. **Charts** > **Variable Settings** を選択します。プロジェクト フォルダに含まれる「BCP Configuration Files」フォルダから *BCP.ini* ファイルを読み出します。
7. 送信ボタンをクリックして、I²C コマンドを CY8C20xx6 コントローラーに送信して、下図に示すように最下部のウインドウのセンサーCp 値を 16 進法で取得します。





UCC 基板のプログラミング

以下のスクリーンショットに示すように#define UART ステートメントに対してコメントを削除して、プロジェクトを生成し、UCC ボードをプログラムします。

```
74
75 /* Comment this while using I2C;Uncomment this while using UART */
76   #define UART
77
```

ハイパーターミナルによるセンサー データの観察

1. Miniprog1 あるいは Miniprog3 を使ってボードに電源を供給します。
2. コンピューターのデスクトップで、**Start** > **All Programs** > **Accessories > Communication > HyperTerminal** を選択します。
3. 接続名を入力します。
4. **OK** をクリックします。



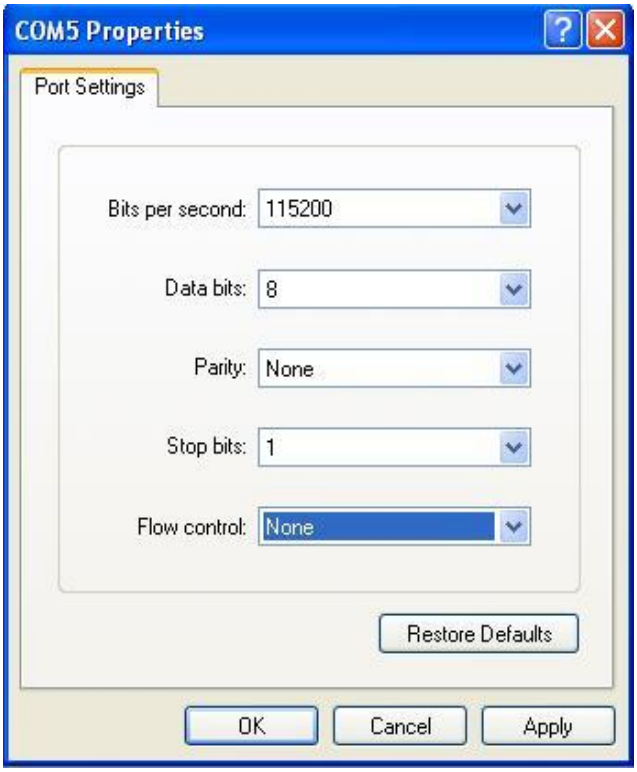
5. 「**Connect** using」 ドロップリストを使用して PSoC からのデータ受信に使用されるシリアル ポートを選択します。
6. **OK** をクリックします。



7. 「**COM Properties**」ダイアログ ボックスでは、以下のパラメーターを設定してください。

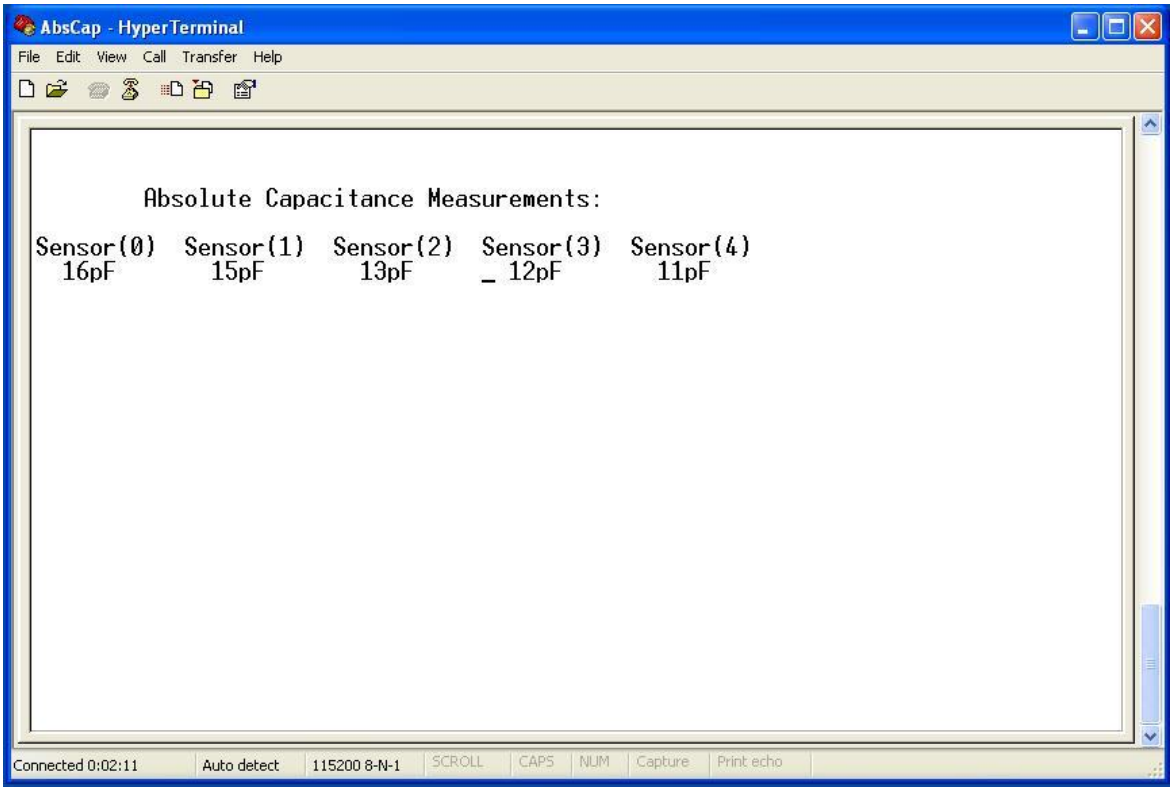
- Bits per second = 115200
- Data bits = 8
- Parity = None
- Stop bits = 1
- Flow control = None

8. **OK** をクリックします。ハイパーターミナルは、指定された COM ポートに接続されており、使える状態にあります。



9. UCC の電源をオンにしてから、リセットします。

結果がハイパーターミナルのウィンドウに表示されます。



サンプルコード 5　絶対センサー静電容量の測定 (CY8C21x34/B CapSense コントローラー使用)

5.1　プロジェクト名

CE_5_MeasuringAbsSensorCap_withCY8C21x34

5.2　概要

このサンプルコードでは、CY8C21x34/B CapSense®コントローラーを使用して、5 個のセンサーの絶対静電容量を計算し、Windows のハイパー ターミナル ソフトウェアまたはブリッジ コントロール パネルを搭載したコンピューターで結果を表示する方法を示します。絶対静電容量は 1pF 単位で測定されます。

5.3　ハードウェアの準備

5.3.1　要求事項

- CY3280-21x34 ユニバーサル CapSense コントローラー
- CY3280-SLM ユニバーサル CapSense リニア スライダー モジュール
- RS232 レベル トランスレータ モジュール
- CY3240-I2USB ブリッジ または CY8CKIT-002 Miniprog3
- CY3217-MiniProg1 Programmer または CY8CKIT-002 Miniprog3
- USB Aーミニ B ケーブル
- Windows XP 以降のオペレーティング システムを実行する PC

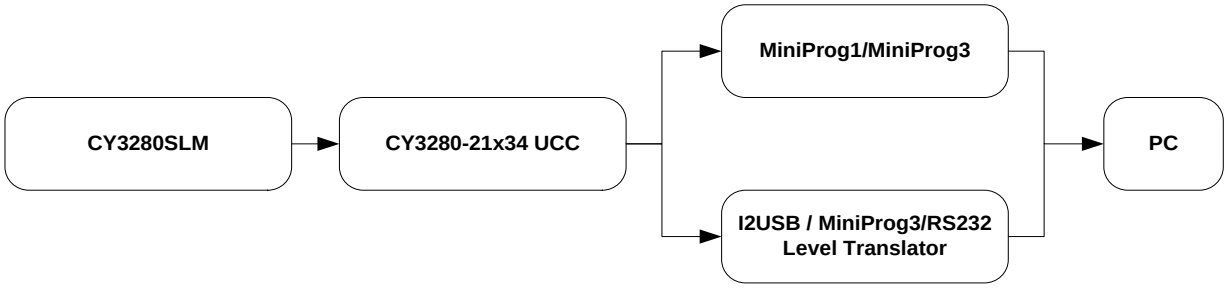
5.3.2　アセンブリ

図 5-1 は、ハードウェアの準備を示します。CY3280-21x34 UCC キットを CY3280-SLM モジュールに 22x2_RA_Receptacle 経由で接続します。MiniProg1/MiniProg3 をヘッダーJ3 に接続してから、キットをプログラミングし、電力を供給します。

ブリッジ コントロール パネル ソフトウェアは、I2C-USB ブリッジまたは MiniProg3 で PC のセンサーCp を表示するために使用されます。

ハイパーターミナルは RS232 通信プロトコルで PC のデータを表示するために使用されます。

図 5-1. ハードウェア準備のブロック図



5.3.3 ボードの設定

以下のハードウェアを接続します。

- CY3280-SLM ドーターカードのヘッダーJ1 を CY3280-21x34UCC ボード上の 22x2_RA_Receptacle (コネクタ P2) に接続します。
- ヘッダーJ1 にジャンパーを嵌め、UCC ボードの V_{CC} ピンと 5 V を接続させます。この設定により、ISSP コネクタから CapSense コントローラーに電力を供給できます。
- ヘッダーJ4 にジャンパーを嵌め、UCC ボードの XRES ピンと XRES/INT ピン (ピン 1 と 2) を接続させます。この設定では、CapSense コントローラーの XRES ピンが ISSP コネクタ J3 のピン 3 に接続されます。
- ヘッダーJ2 にジャンパーを嵌め、CY3280-SLM ボードの GND ピンと SHILED ピン (ピン 2 と 3) を接続させます。この設定により、CY3280-SLM ボード上のハッチパターンがグランド接続されます。
- MiniProg1/MiniProg3 を J3 ヘッダー上の CY3280-21x34 UCC に接続します。
- センサーCp が I2C 通信プロトコルによってブリッジ コントロール パネルで表示される場合、CY3280-21x34 のジャンパ J3 を I2C-USB ブリッジ/MiniProg3 に、および USB A ～ ミニ B ケーブルによってブリッジの残りの端をコンピューターに接続させます。ブリッジ コントロール パネルのトグルパワー ボタンをクリックしてデバイスに電源を供給します。
- センサーCp が UART によりハイパーターミナルで表示される場合、RS232 レベル トランスレータ ボード上の TX ピンを UCC の P1[7]に接続させます。レベル トランスレータの Vcc ピンと GND ピンを、UCC の Vcc ピンと GND ピンにそれぞれ接続します。RS232 レベル トランスレータの好例は、MAX232 シリアル レベル コンバータです。
- シリアル ケーブルを使用して、レベル トランスレータ ボードの RS232 ポートをコンピューターに接続します。

5.4 回路図

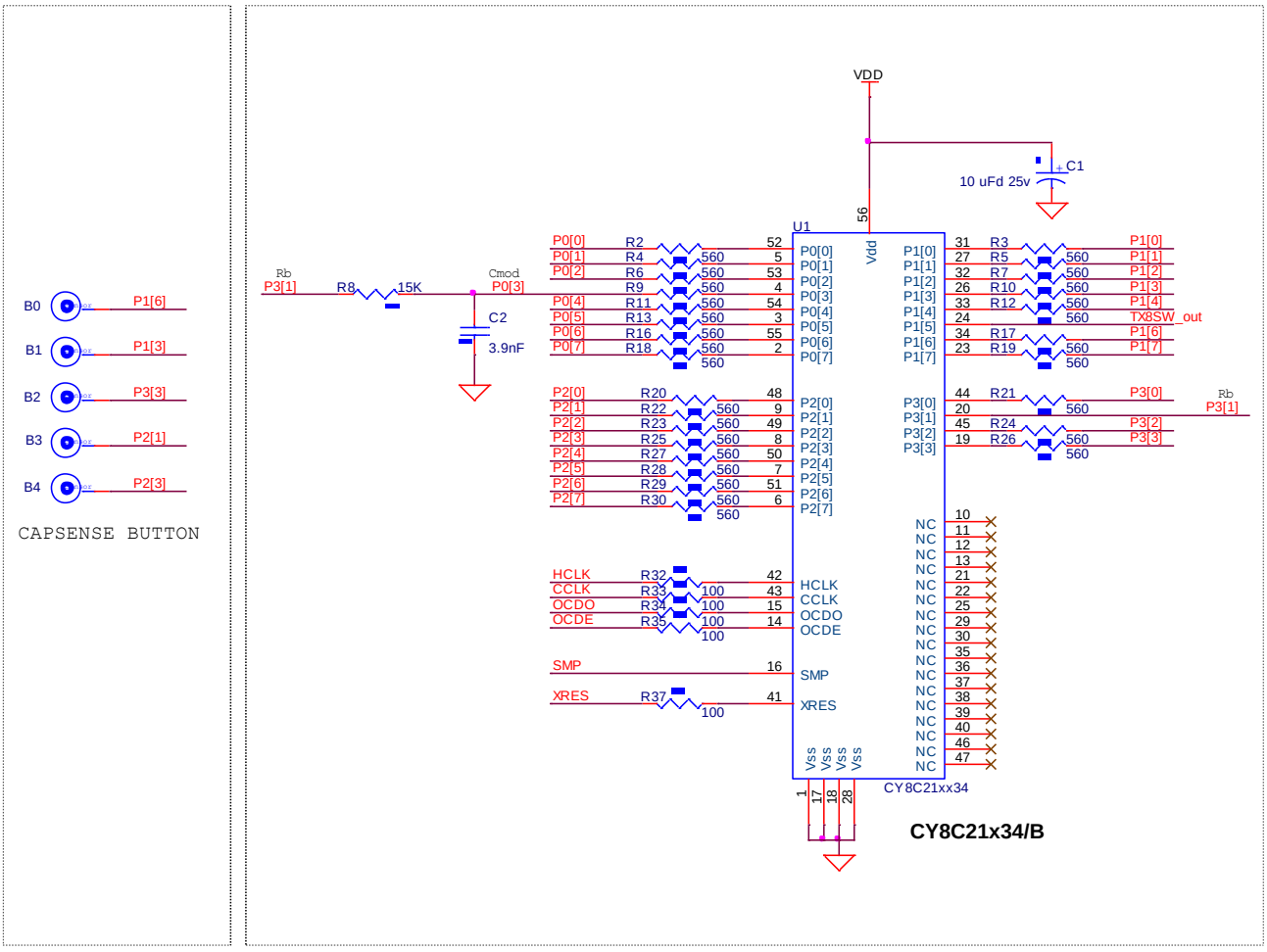


表 5-1.サンプルコード 5 のピン割り当て

対象	ピン
ボタン 0	P1[6]
ボタン 1	P1[3]
ボタン 2	P3[3]
ボタン 3	P2[1]
C_{MOD}	P0[3]
Rb	P3[1]

5.5 ソフトウェアの準備

5.5.1 ツール

- PSoC® Designer (バージョン 5.2 以降)
- PSoC Programmer (バージョン 3.13 以降)
- Bridge Control Panel (ブリッジ コントロール パネル)
- Windows ハイパーターミナル

5.5.2 ユーザーモジュール

以下の表は、このサンプルコードで使用されるユーザーモジュール(UM)、および各 UM が占有するハードウェア リソースを一覧にします。

ユーザー モジュール	ハードウェア リソース
CSD	CapSense ブロック、タイマー1 (デフォルト)
EzI2C	I2C ブロック
TX8SW	占有するブロックなし (ソフトウェア実装)

5.5.3 ユーザーモジュール パラメーター、グローバル リソース

プロジェクトの `ReadMe.txt` ファイルでは、このサンプルコードで使用されるユーザーモジュールのパラメーター設定について説明しています。また、グローバル リソースのリストが含まれています。

5.6 動作

このプログラムでは、センサーの raw カウントの計算に以下の公式を使用します。

$$\text{Raw Counts} = C_P \,.\, \text{Sensitivity}$$

ここでは、C_P は、センサー静電容量です。

このように、このプログラムでは、raw カウントと感度の比率を取得して、各センサーの絶対静電容量を計算します。

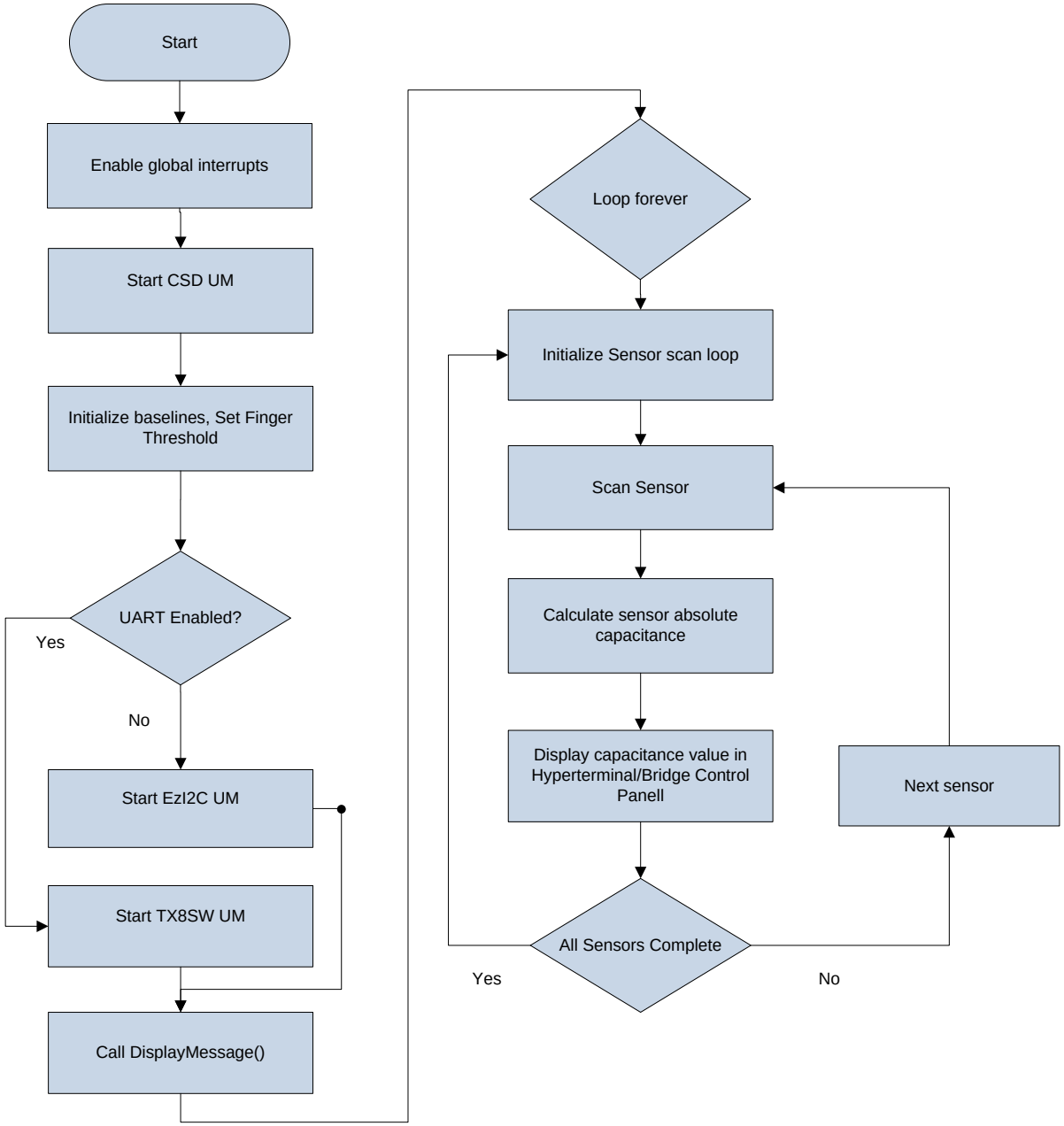
このプロジェクトには、`CY8C21x34/B_CalculSensitivity.xls` というファイルが含まれています。このファイルには、感度の計算式が含まれています。最も正確な結果を生じた基準値とプリスケーラのパラメーターの設定は、

- Ref Value = 5
- Prescaler = 5

このプログラムでは、以下の手順を使用して画面に絶対静電容量を表示します。

1. この設定では、計算された感度は 465 カウント/pF と等しくなります。*CY8C21x34/B_CalculSensitivity.xls* ファイルを参照してください。
   .xls ファイルはこのプロジェクトの一部であり、プロジェクト フォルダに保管されています。
2. グローバル割り込みを有効にしてから、ユーザーモジュールを開始します。
3. 各センサーを順番にスキャンし、配列 CSD_waSnsResult[] から raw カウントを取得します。
4. raw カウントと感度の比率を取得して、絶対静電容量を計算します。
5. ユーザーの選択に応じてブリッジ コントロール パネルまたはハイパーターミナルを表示するために結果をコンピューターに送信します。

図 5-2. サンプルコード 5 の機能フローチャート



5.7 コードの実行

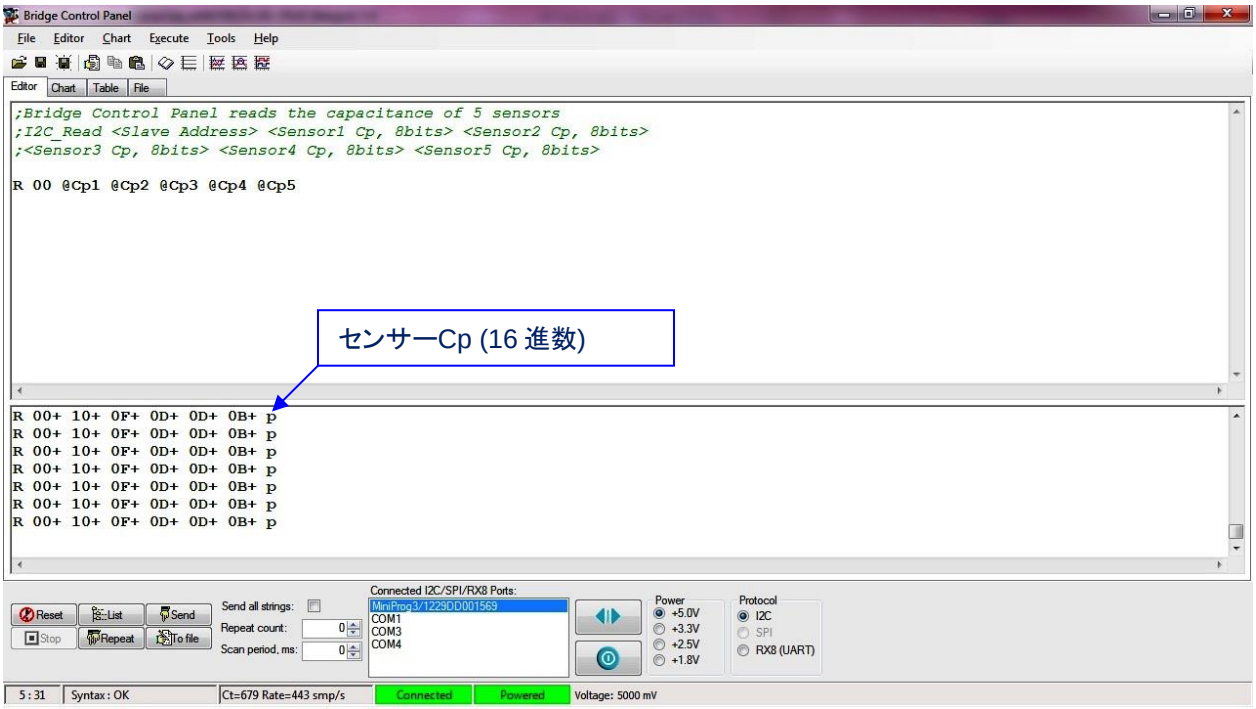
基板のプログラミング

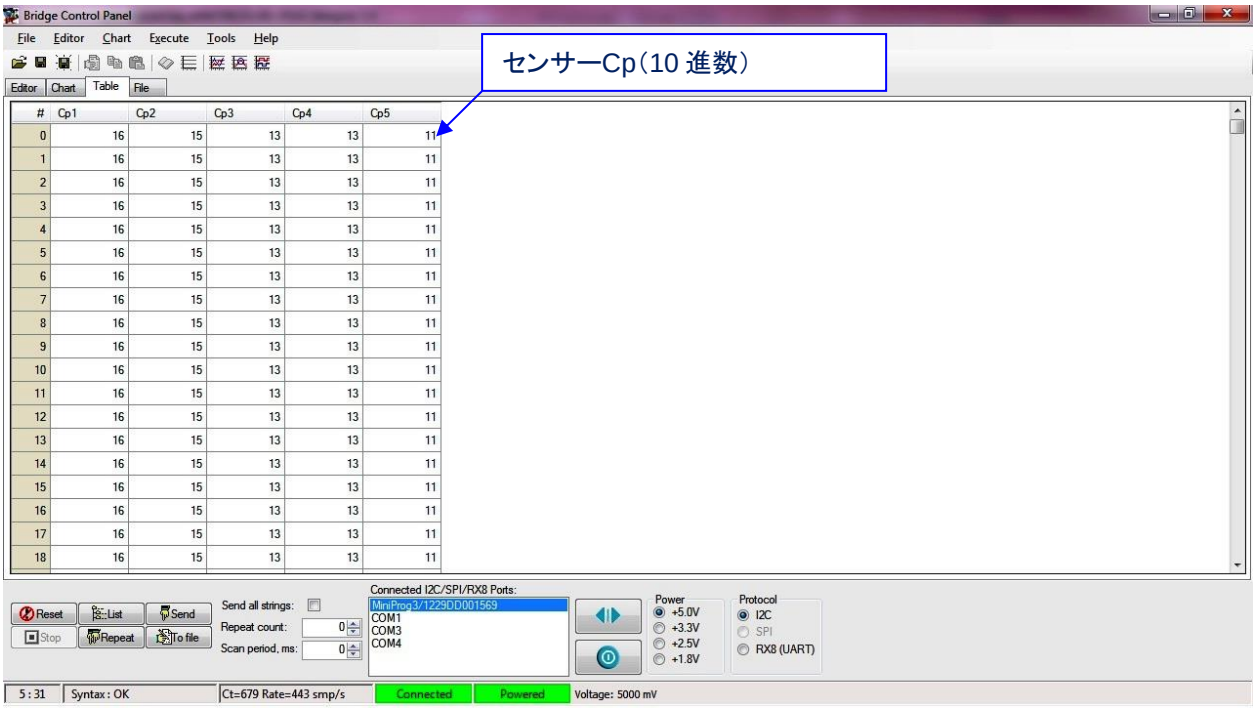
以下のスクリーンショットに示すように *main.c* 内の#define UART ステートメントがコメントアウトされたことを確保し、プロジェクトを生成し、UCC ボードをプログラムします。UCC ボードのプログラミングの方法については、CY3280-21x34 キット ガイドの第 5 章を参照してください。

```
76 /* comment this while using I2C;Uncomment this while using UART */
77
78  //#define UART
79
```

BCP によるセンサー データの観察

1. CY3280-21x34 のジャンパ J3 を I2C-USB ブリッジ/MiniProg3 に、および USB A ～ ミニ B ケーブルによって I2C-USB ブリッジ/MiniProg3 の残りの端をコンピューターに接続させます。
2. コンピューターのデスクトップで、Start > **All Programs** > Cypress > **Bridge Control Panel** (バージョン番号) > **Bridge Control Panel** (バージョン番号) を選択します。
3. ポート選択ウィンドウからデバイスを選択します。
4. CY3280-21X34 UCC キット ボードに 5V で電力を供給します。
5. Bridge Control Panel ツールで **File** > **Open** を選択します。サンプルコード プロジェクト フォルダに含まれる「BCP Configuration Files」フォルダから BCP.ini ファイルを読み出します。
6. **Charts** > **Variable Settings** を選択します。プロジェクト フォルダに含まれる「BCP Configuration Files」フォルダから *BCP.ini* ファイルを読み出します。
7. 送信ボタンをクリックして、I²C コマンドを CY8C21x34 コントローラーに送信して、下のスクリーンショットに示すように最下部のウインドウのセンサーCP 値を 16 進法で取得します。





ボードのプログラミング

main.c 内の#define UART ステートメントに対してコメントを解除し、ボードをプログラムします。

```
/* comment this while using I2C;Uncomment this while using UART */

 #define UART

```

HyperTerminal を使ってセンサー データを監視します。

1. Miniprog1 あるいは Miniprog3 を使ってボードに電源を供給します。
2. コンピューターのデスクトップで、Start > All Programs > Accessories > Communication > HyperTerminal を選択します。
3. 接続名を入力します。
4. **OK** ボタンをクリックします。

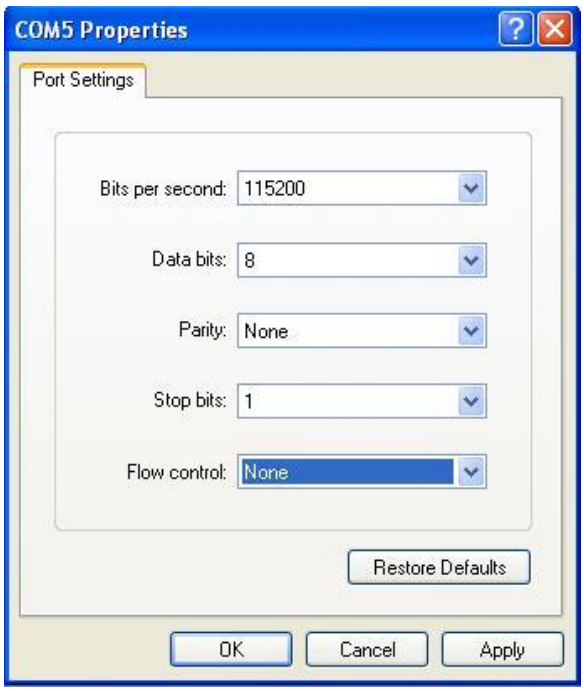


5. 「Connect using」ドロップリストを使用して、SoC からデータを受領するのに使用するシリアル ポートを選択します。この設定は、PC のコンフィギュレーションにより異なります。この例では、COM5 を使用します。
6. 「OK」をクリックします。



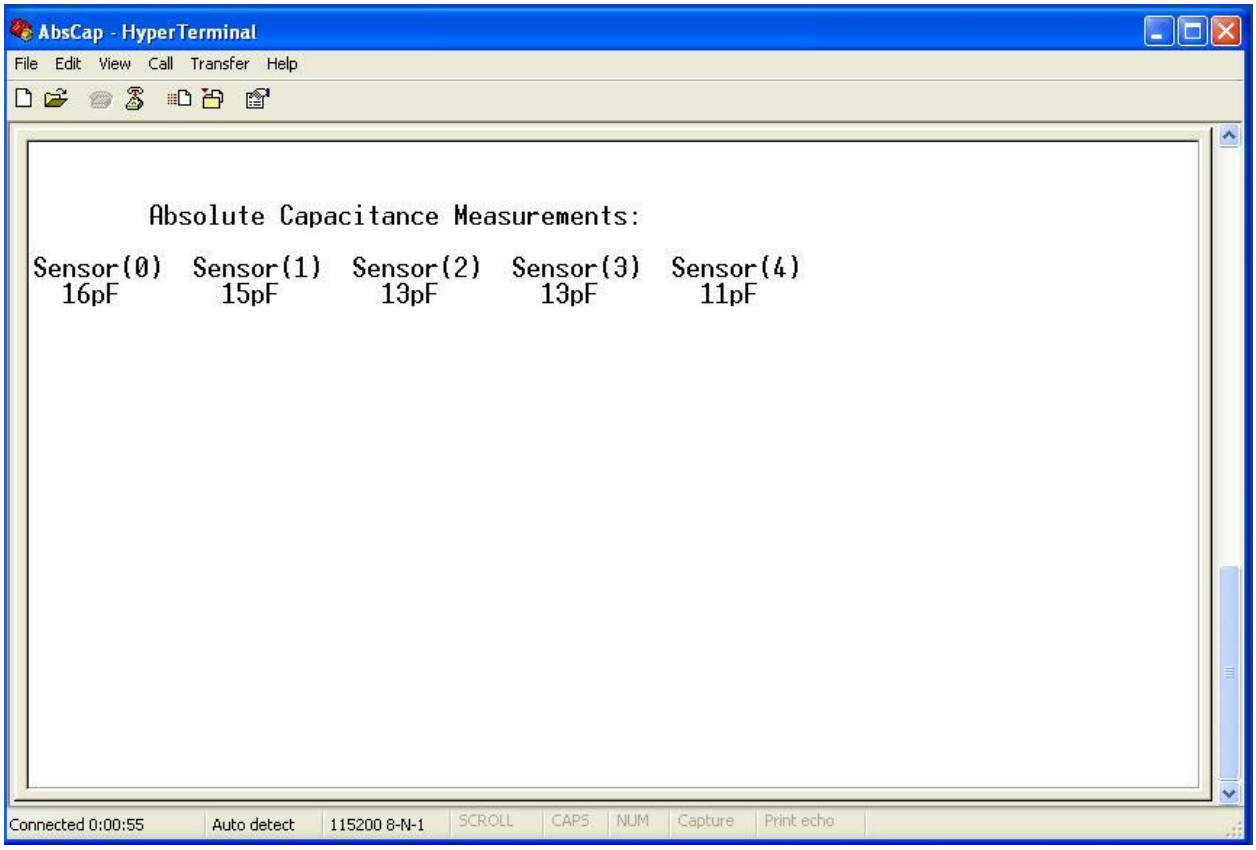
7. COM Properties ダイアログボックスでは、以下のパラメーターを設定します。
   - Bits per second = 115200
   - Data bits = 8
   - Parity = None
   - Stop bits = 1
   - Flow control = None
8. **OK** をクリックします。

   ハイパーターミナルは、指定された COM ポートに接続されており、使える状態にあります。



9. UCC の電源をオンにしてから、リセットします。

   結果がハイパーターミナルのウィンドウに表示されます。



サンプルコード 6　絶対センサー静電容量の測定 (CY8C20x34 CapSense コントローラー使用)

6.1　プロジェクト名

CE_6_MeasuringAbsSensorCap_withCY8C20x34

6.2　概要

このサンプルコードは、CY3280-20x34 CapSense®コントローラーを使用して 5 個のセンサーの絶対静電容量を計算します。絶対静電容量は 1pF 単位で測定されます。

6.3　ハードウェアの準備

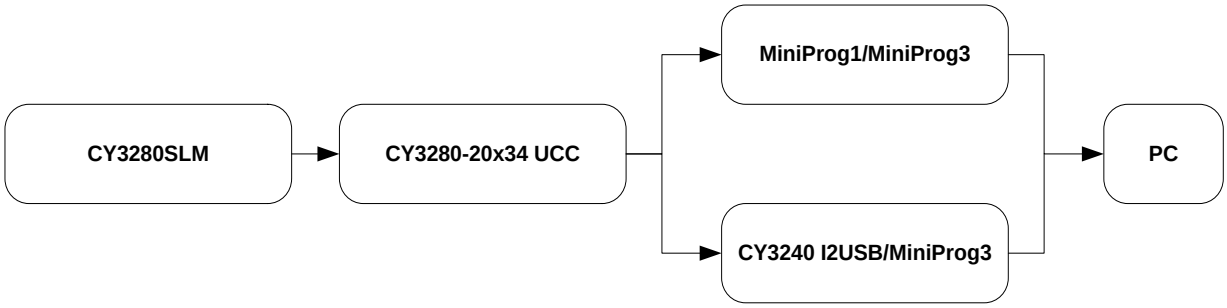
6.3.1　要求事項

- CY3280-20x34 ユニバーサル CapSense コントローラー
- CY3280-SLM ユニバーサル CapSense リニア スライダー モジュール
- CY3217 – MiniProg1 Programmer または CY8CKIT-002 Miniprog3
- CY3240 – I2USB ブリッジ または CY8CKIT-002 Miniprog3
- USB Aーミニ B ケーブル
- Windows XP 以降のオペレーティング システムを実行する PC
- マルチメーター

6.3.2　組み立て

図 6-1 は、ハードウェアの準備を示します。CY3280-20x34 UCC キットを CY3280-SLM モジュールに 22x2_RA_Receptacle 経由で接続します。MiniProg1/Miniprog3 をヘッダーJ3 に接続してから、キットをプログラミングし、電力を供給します。

図 6-1. ハードウェア セットアップのブロック図

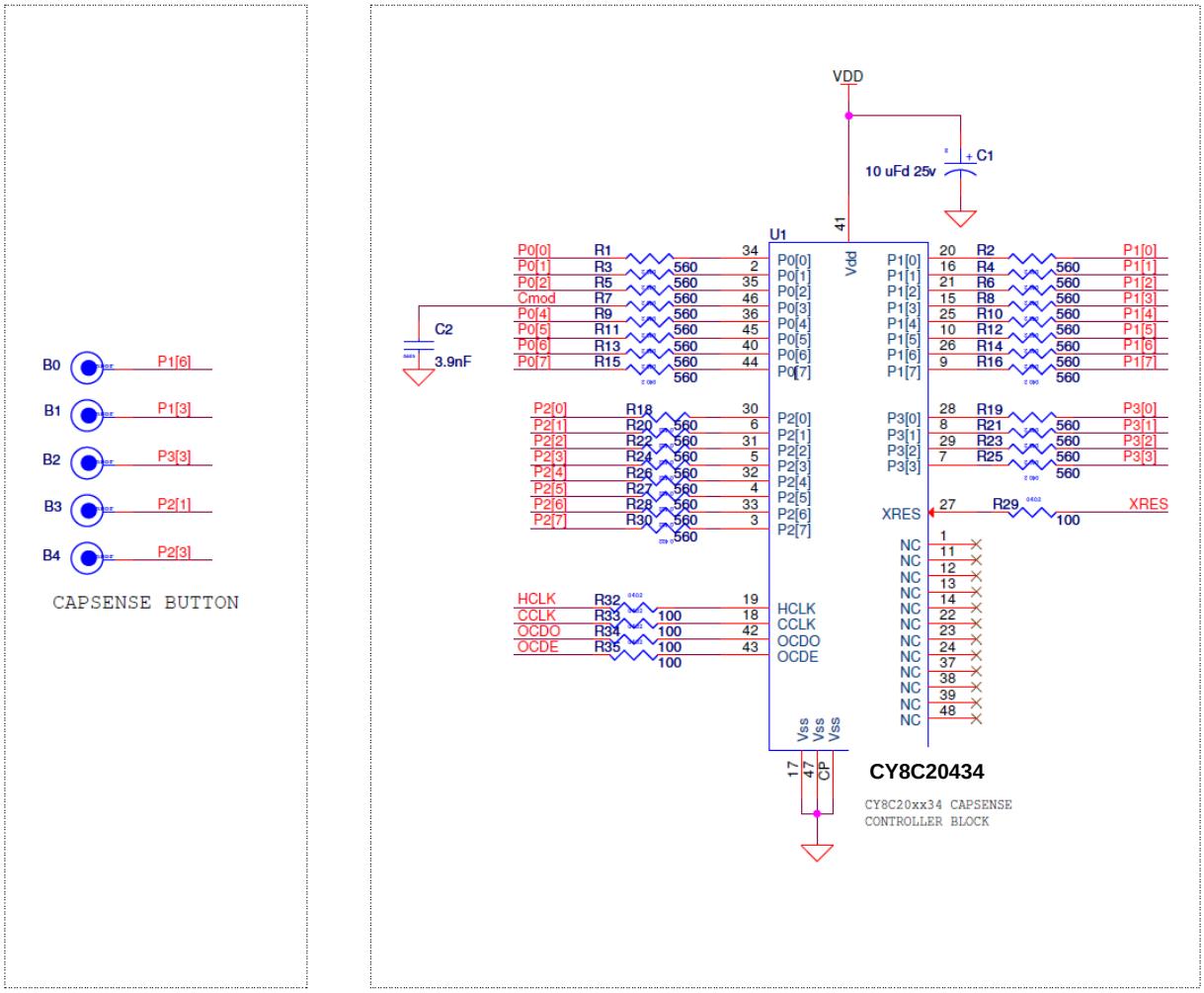


6.3.3 ボードの設定

以下のハードウェアを接続します。

- CY3280-SLM ドーターカードのヘッダーJ1 を CY3280-20x34UCC ボード上の 22x2_RA_Receptacle (コネクタ P2) に接続します。
- ヘッダーJ1 にジャンパーを嵌め、UCC ボードの V_{CC} ピンと 5 V を接続させます。この設定により、ISSP コネクタから CapSense コントローラーに電力を供給できます。
- ヘッダーJ4 にジャンパーを嵌め、UCC ボードの XRES ピンと XRES/INT ピン(ピン 1 と 2) を接続させます。この設定では、CapSense コントローラーの XRES ピンが ISSP コネクタ J3 のピン 3 に接続されます。
- ヘッダーJ2 にジャンパーを嵌め、CY3280-SLM ボードの GND ピンと SHILED ピン (ピン 2 と 3) を接続させます。この設定により、CY3280-SLM ボード上のハッチパターンがグランド接続されます。
- MiniProg1/Minirpog3 を UCC ボードの J3 上の ISSP ヘッダーに接続します。この接続は、コードで生成される hex ファイルで UCC をプログラミングするときのみ要求されます。この接続は、CapSense データが Bridge Control Panel ソフトウェアで読み出される時、I2USB ブリッジと置き換える必要があります。
- USB Aーミニ B ケーブルを使用して、MiniProg1/Miniprog3 (または I2USB ブリッジ) の他の端を PC に接続します。

6.4 回路図



変調コンデンサ (C_{MOD}) は、P0[3] に接続された 3.9nF のコンデンサです。560Ω の抵抗は、RF 干渉を軽減するため、各 CapSense ボタンに直列に接続されます。

表 6-1. サンプルコード 6 のピン割り当て

ボタン	ピン
0	P1[6]
1	P1[3]
2	P3[3]
3	P2[1]
4	P2[3]

6.5 ソフトウェアの準備

6.5.1 ツール

- PSoC® Designer (バージョン 5.2 以降)
- PSoC Programmer (バージョン 3.13 以降)
- Bridge Control Panel (ブリッジ コントロール パネル)

6.5.2 ユーザーモジュール

以下の表は、このサンプルコードで使用されるユーザーモジュール(UM)、および各 UM が占有するハードウェア リソースを一覧にします。

ユーザーモジュール	ハードウェア リソース
CSA_EMC	CapSense ブロック、コンパレータ (デフォルト)
EzI2Cs	I²C/SPI

6.5.3 ユーザーモジュール パラメーター、グローバル リソース

プロジェクトの `ReadMe.txt` ファイルでは、このサンプルコードで使用されるユーザーモジュールのパラメーター設定について説明しています。また、グローバル リソースのリストが含まれています。

6.6 動作

main.c ファイルには、2 つの機能があります。1 つは各センサーの IDAC コードを計算する機能です。もう 1 つは、IDAC 電流を直接測定できるポート ピンに対応の電流を駆動します。各センサーの絶対静電容量は、以下の方程式で計算します。

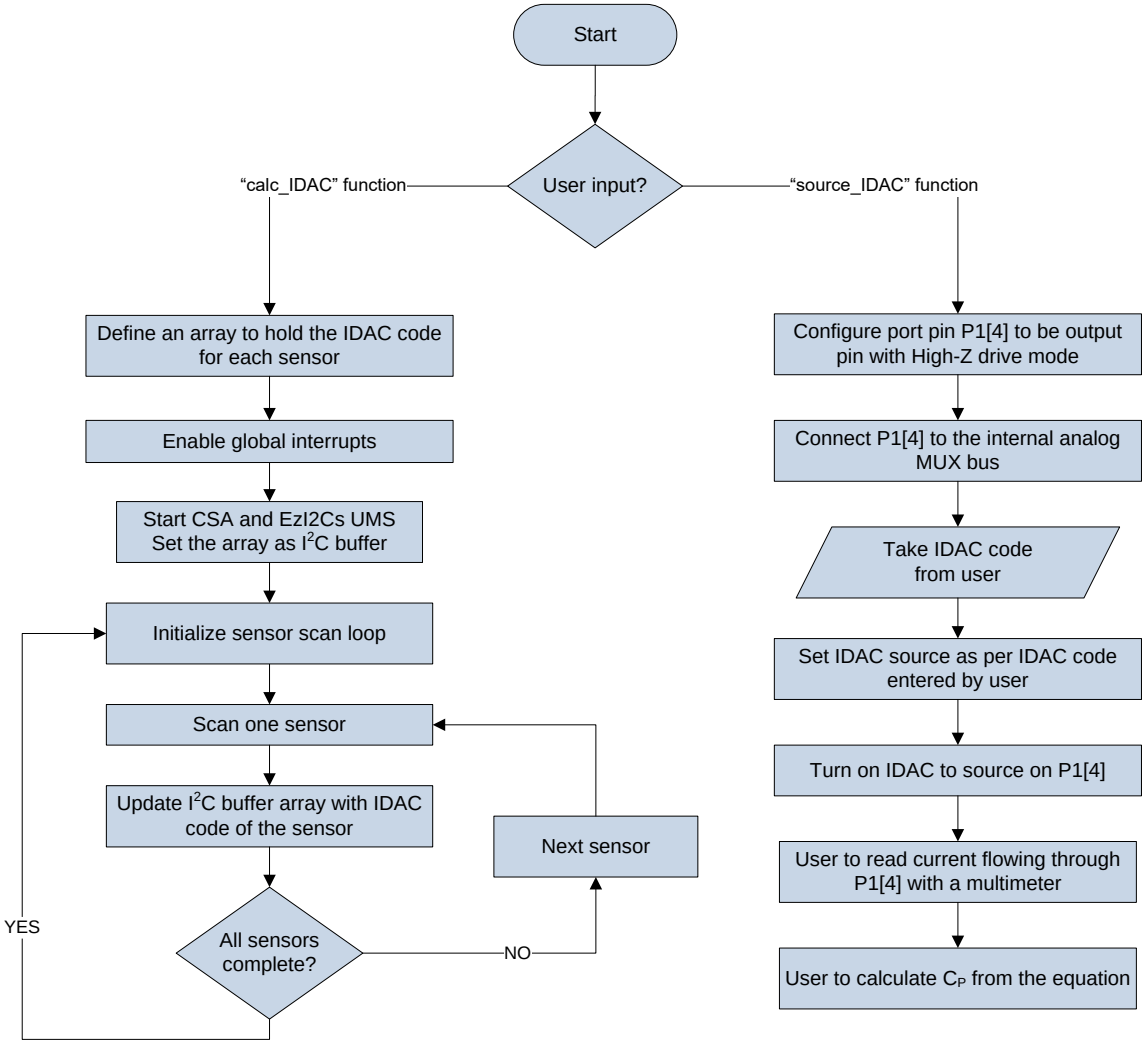
$$C_P = \frac{I_{measured}}{\left(IMO/8\right) \times 1.3}$$

ここでは、C_P はセンサー静電容量です。

プログラムは、以下の手順で処理します。

- 2 つの関数 calc_IDAC および source_IDAC を定義してから、ユーザー入力に応じてどちらかを呼び出します。
- 最初の関数 calc_IDAC は、以下の動作を実行します。
  - 配列 baIDAC を定義し、センサー番号に従って各 CapSense センサーの IDAC コードを保管します。
  - 配列を I²C 読み出しバッファとして設定します。
  - グローバル割り込みを有効にしてから、ユーザーモジュールを起動します。
  - 無限ループで、各センサーを順番にスキャンし、配列に保管されている IDAC コードを取得します。I²C バッファも更新します。
- 2 つ目の関数 source_IDAC は、以下の動作を実行します。
  - ポートピン P1[4]を High-Z ドライブ設定の出力として設定します。
  - ピン P1[4]を内部アナログ MUX バスに接続します。
  - 入力された IDAC コードを読み出します。
  - ピン P1[4]に対応する電流を供給します。

図 6-2. サンプルコード 6 の機能フローチャート

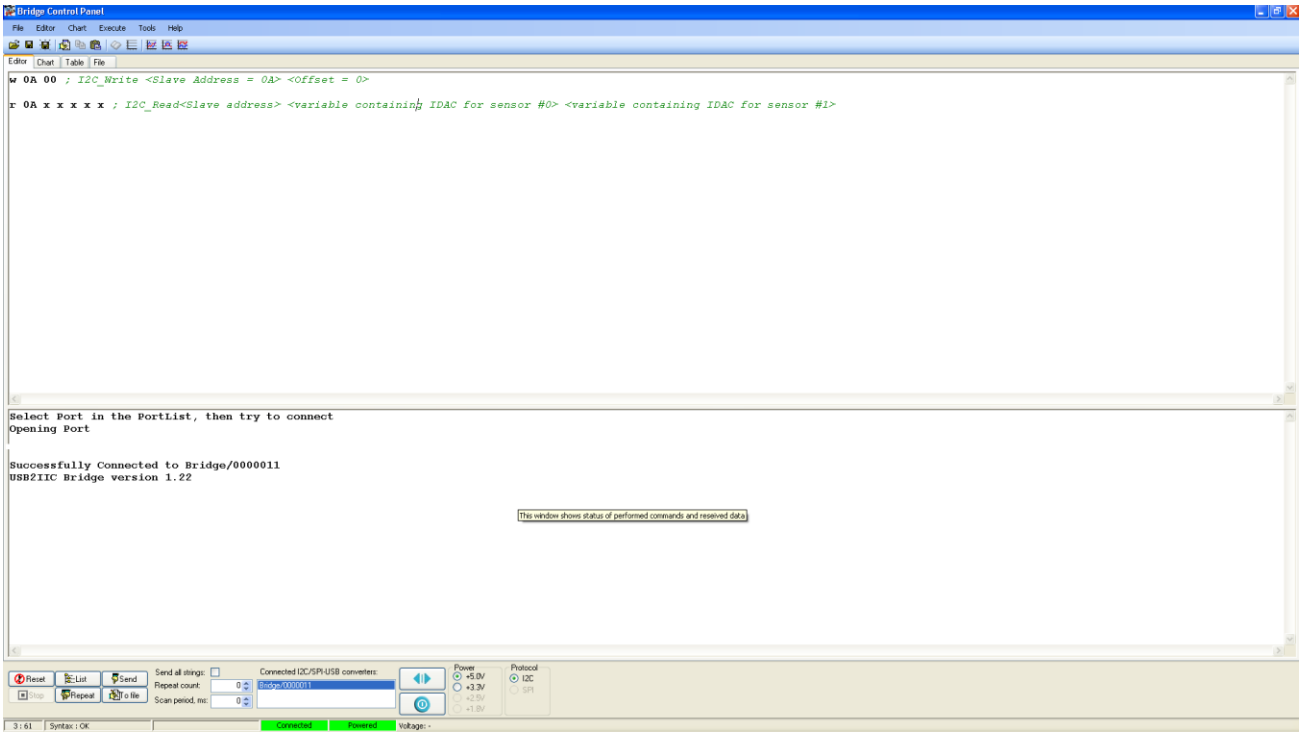


6.7 サンプルコードの実行

ボードにサンプルプロジェクトをプログラムしてから、この手順を使用してサンプルコードを実行します。UCC 基板のプログラミングの方法については、CY3280-20x34 キット ガイドの第 5 章を参照してください。

1. I2USB ブリッジと USB A－ミニ B ケーブルを使用して、コンピュータを CY3280-20x34 ユニバーサル CapSense コントローラー ボードの ISSP コネクタ J3 に接続させます。
2. コンピューターのデスクトップで、**Start** > **All Programs** > **Cypress** > **Bridge Control Panel** (バージョン番号) > **Bridge Control Panel** (バージョン番号) を選択します。
   Bridge Control Panel ツールは、PSoC Programmer のインストール コンポーネントの 1 つです。
3. ポート選択ウィンドウからデバイスを選択します。
4. CY3280-20X34 CapSense コントローラー ボードに 5V で電力を供給します。
5. Bridge Control Panel ツールで、 **Tools** > **Protocol Configuration** を選択します。**I2C** タブで、**I2C Speed** を 100kHz に設定します。
6. **File** > **Open** を選択します。サンプルコード プロジェクト フォルダに含まれる「BCP Configuration Files」フォルダから `BCP.iic` ファイルを読み出します。

図 6-3. ステータス ウィンドウでスレーブ アドレスを表示



7. [**Enter**] を押して、Bridge Control Panel の 2 番目の指示を実行します。
   ステータス ウィンドウに、センサー全部の IDAC コードが表示されます。
8. 5 個のセンサーの IDAC コードが表示されたら、プロジェクトを開き、ステートメント *source_IDAC(),*を非コメント化し、ステートメント *calc_IDAC()*をコメント化して、ポートピン P1[4]に対応する電流を供給します。
9. 関数 *source_IDAC()*で、適切な 16 進数を IDAC_D 変数に割り当てて、IDAC コードをいずれかのセンサーの Bridge Control Panel で取得したセンサーに設定します。
10. PSoC Designer を使用して、このプロジェクトをビルドします。 **Build** > **Generate** または **CE_6_MeasuringAbsSensorCap_withCY8C20x34** を選択します。ビルド中にエラーまたは警告が表示されないことを確認します。
11. 生成された hex ファイルで UCC をプログラミングするには、**Program** > **Program Part** を選択します。
12. Program Part ウィンドウで、MiniProg1/Miniprog3 が接続されていることを確認します。接続されていない場合は、**Connect** ボタンをクリックします。
13. **Program** ボタンをクリックします。
14. UCC から CY3280-ボードの接続を切断します。
15. パワー ボタンをクリックして、MiniProg1/Miniprog3 から UCC に電力を供給します。
16. ピンとグランドの間に電流計を挿入し、P1[4]から流れる電流 ($I_{MEASURED}$) を測定します。
17. 計算式を使用して、C_P を計算します。

$$C_P = \frac{I_{measured}}{\left(IMO/8\right) \times 1.3}$$

    ここで、IMO は、システム クロック ソースです。このプロジェクトでは、12MHz に設定します。
18. ステップ 9 から 17 までを繰り返して、他のセンサーの C_P を取得します。

表 6-2 では、CapSense センサーの C_P 値の他に、Bridge Control Panel を介して取得した IDAC コード、およびピンで供給される IDAC 電流が一覧にします。

表 6-2. 測定 C_P

CapSense センサー	IDAC コード	ポートピン (µA) に供給される IDAC 電流	C_P 測定値 (pF)
P1[6]	0A	28.0	14.40
P1[3]	09	25.3	12.98
P3[3]	08	22.6	11.58
P2[1]	07	19.3	9.90
P2[3]	07	19.3	9.90

サンプルコード 7　CSD の調整 (CY8C20xx6A CapSense コントローラー使用)

7.1　プロジェクト名

CE_7_TuningCSD_withCY8C20xx6A

7.2　概要

このサンプルコードでは、CY8C20xx6A デバイスの CapSense®を調整する方法を示しています。CapSense UM を調整するには、CapSense データをグラフ形式で表示できなければなりません。このサンプルコードでは、EzI2C UM を使用して必要な CapSense パラメーターを I²C バスと I2USB ブリッジに転送します。PC で Bridge Control Panel を使用して、このデータを表示します。

7.3　ハードウェアのセットアップ

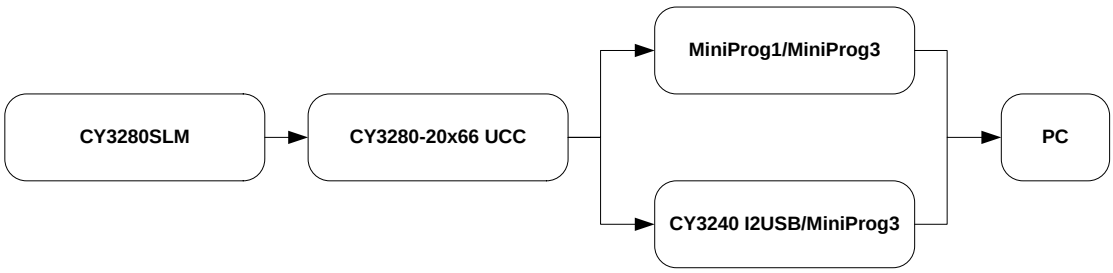
7.3.1　要求事項

- CY3280-20x66 ユニバーサル CapSense コントローラー ボード
- CY3280-SLM ユニバーサル CapSense リニア スライダー モジュール
- CY3240-I2USB ブリッジまたは CY8CKIT-002 Miniprog3
- CY3217-MiniProg1 プログラマ キットまたは CY8CKIT-002 Miniprog3
- USB Aーミニ B ケーブル
- Windows XP 以降のオペレーティング システムを実行する PC

7.3.2　組み立て

図 7-1 は、ハードウェアの準備を示します。CY3280-20x66 UCC キットを CY3280-SLM モジュールに 22x2_RA_Receptacle 経由で接続します。MiniProg1/MiniProg3 または I2USB ブリッジ/MiniProg3 をキットの ISSP ヘッダーに接続します。ハードウェア準備では、プログラミングに MiniProg1/MiniProg3 を使用し、PC にデータを送信するには I2USB ブリッジ/Miniprog3 を使用します。これらは、USB ケーブルで PC に接続します。

図 7-1. ハードウェア セットアップのブロック図

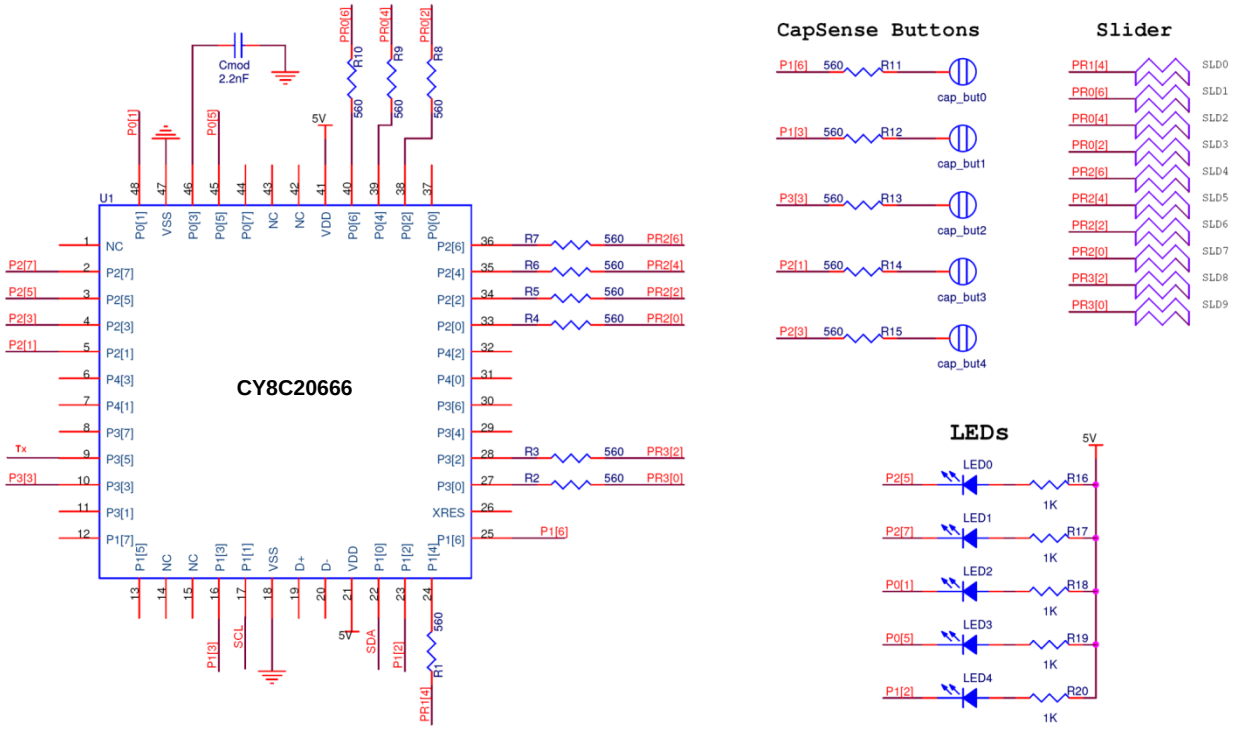


7.3.3 ボードの設定

以下のハードウェアを接続します。

- CY3280-SLM ドーターカードのヘッダーJ1 を CY3280-20x66 UCC ボード上の 22x2_RA_Receptacle (コネクタ P2) に接続します。
- ヘッダーJ7 にジャンパーを嵌め、UCC ボードの VCC ピンと VCC_PROG ピンを接続させます。この設定により、ISSP コネクタから CapSense コントローラーに電力を供給できます。
- ヘッダーJ4 にジャンパーを嵌め、UCC ボードの XRES ピンと XRES/INT ピン(ピン 1 と 2) を接続させます。この設定では、CapSense コントローラーの XRES ピンが ISSP コネクタ J3 のピン 3 に接続されます。
- ヘッダーJ2 にジャンパーを嵌め、CY3280-SLM ボードの GND ピンと SHILED ピン (ピン 2 と 3) を接続させます。この設定により、CY3280-SLM ボード上のハッチパターンがグランド接続されます。
- MiniProg1/Miniprog3 を UCC ボードの J3 上の ISSP ヘッダーに接続します。この接続は、コードで生成される hex ファイルで UCC をプログラミングするときのみ要求されます。この接続は、CapSense データが Bridge Control Panel ソフトウェアで読み出される時、I2USB ブリッジと置き換える必要があります。
- USB A－ミニ B ケーブルを使用して、MiniProg1/Miniprog3 (または I2USB ブリッジ) の他の端を接続します。

7.4 回路図



変調コンデンサ (C_{MOD}) は、P0[3] に接続された 2.2nF のコンデンサです。560Ω の抵抗は、RF 干渉を軽減するため、各 CapSense ボタンに直列に接続されます。LED は 1kΩ の直列抵抗に接続しアクティブローの設定です。全部で、LED が 5 個、ボタンが 5 個、セグメントが 10 あるスライダーが 1 個あります。

表 7-1. LED、ボタン、スライダー セグメントのピン割り当て

LED	ボタン	スライダー セグメント
LED0 - P2[5]	BTN0 - P1[6]	SLD0 - P1[4]、SLD1 - P0[6]

LED1 - P2[7]	BTN1 - P1[3]	SLD2 - P0[4]、SLD3 - P0[2]
LED2 - P0[1]	BTN2 - P3[3]	SLD4 - P2[6]、SLD5 - P2[4]
LED3 - P0[5]	BTN3 - P2[1]	SLD6 - P2[2]、SLD7 - P2[0]
LED4 - P1[2]	BTN4 - P2[3]	SLD8 - P3[2]、SLD9 - P3[0]

7.5 ソフトウェアの準備

7.5.1 ツール

- PSoC® Designer (バージョン 5.2 以降)
- PSoC Programmer (バージョン 3.14 以降)
- Bridge Control Panel (ブリッジ コントロール パネル)

7.5.2 ユーザーモジュールと配置

以下の表は、このコード表で使用されているユーザーモジュール 、および各ユーザーモジュールが占有するハードウェア リソースを一覧にします。

ユーザーモジュール	配置
CSD	CapSense ブロック、タイマー1 (デフォルト)
EzI2C	I²C/SPI ブロック

7.5.3 ユーザーモジュール パラメーター、グローバル リソース

プロジェクトの `ReadMe.txt` ファイルでは、このサンプルコードで使用されるユーザーモジュールのパラメーター設定について説明しています。また、グローバルリソースのリストが含まれています。

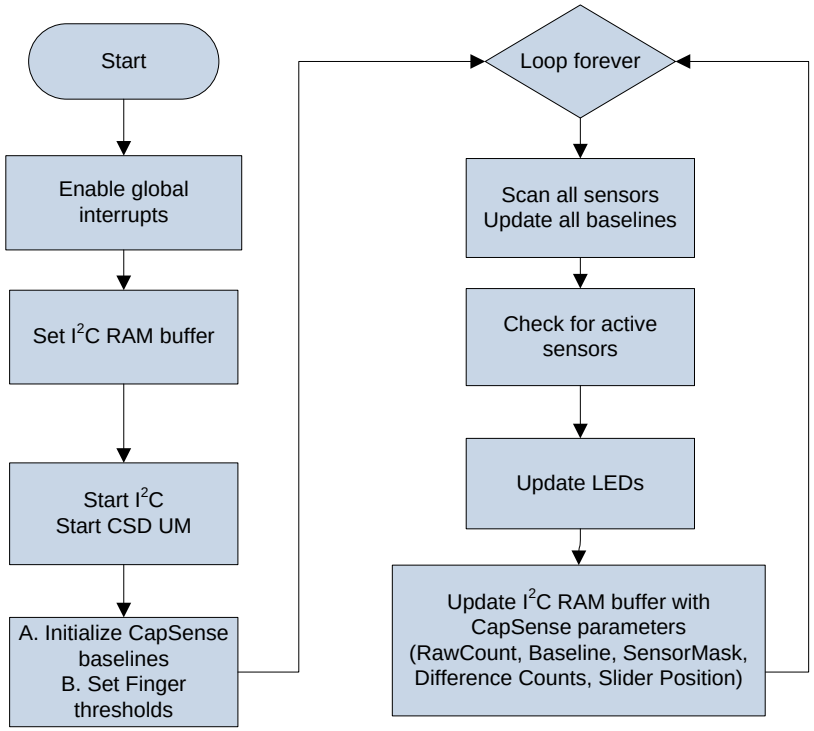
注: サンプルコードは、 CY3280-SLM キットに付属するアクリル製オーバーレイ (1.5mm) 向けに調整しました。オーバーレイがこれより厚い場合は、このサンプルコードで説明しているプロジェクトを調整します。

7.6 動作

リセットすると、ファームウェアは以下の動作を実行します。

- 構造体 MyI2C_Regs を定義してボタン番号、raw カウント、差分カウント、ベースライン、セントロイド位置、所定のボタン番号に対応する CapSense ボタンの状態を保管します。
- グローバル割り込みを有効にしてから、CSD のユーザーモジュールを開始します。
- EzI2Cs UM を開始し、構造体 MyI2C_Regs を I²C RAM バッファとして設定します。
- 無限ループで以下の操作を実行します。
  - すべてのセンサーを継続的にスキャンし、構造体 MyI2C_Regs を raw カウント、差分カウント、ベースライン、スライダーのセントロイド位置、および要求される CapSense ボタンで更新します。I²C マスターは、ボタン番号を EzI2C スレーブの I²C バッファの最初のバイトに書き込み、特定のボタンの CapSense を要求することができます。
  - ファームウェアはボタンが押されるのを検出すると、対応する LED を点灯させます。ボタンを放すと、LED を消灯させます。
  - スライダーに触れると、ファームウェアは LED を点灯させて接触位置を示します。

図 7-2. サンプルコード 7 の機能的フローチャート



7.7 コードの実行

ボードにサンプル プロジェクトをプログラムしてから、この手順を使用してサンプルコードを実行します。UCC ボードのプログラミングの方法については、CY3280-20x66 キット ガイドの第 5 章を参照してください。

1. ボードを 5V で給電するには、MiniProg1/Miniprog3 または CY3280-20x66 UCC キットガイドで説明している電源のいずれかを使用します。
2. CY3280-SLM モジュール ボードでリニア スライダーに触れます。
   CY3280-SLM モジュール ボードの対応する LED が点灯します。
3. ボタンに触れます。
   CY3280-SLM モジュール ボードの対応する LED が点灯します。同時に複数のボタンが機能します。リニア スライダーとボタンも同時に使用できます。

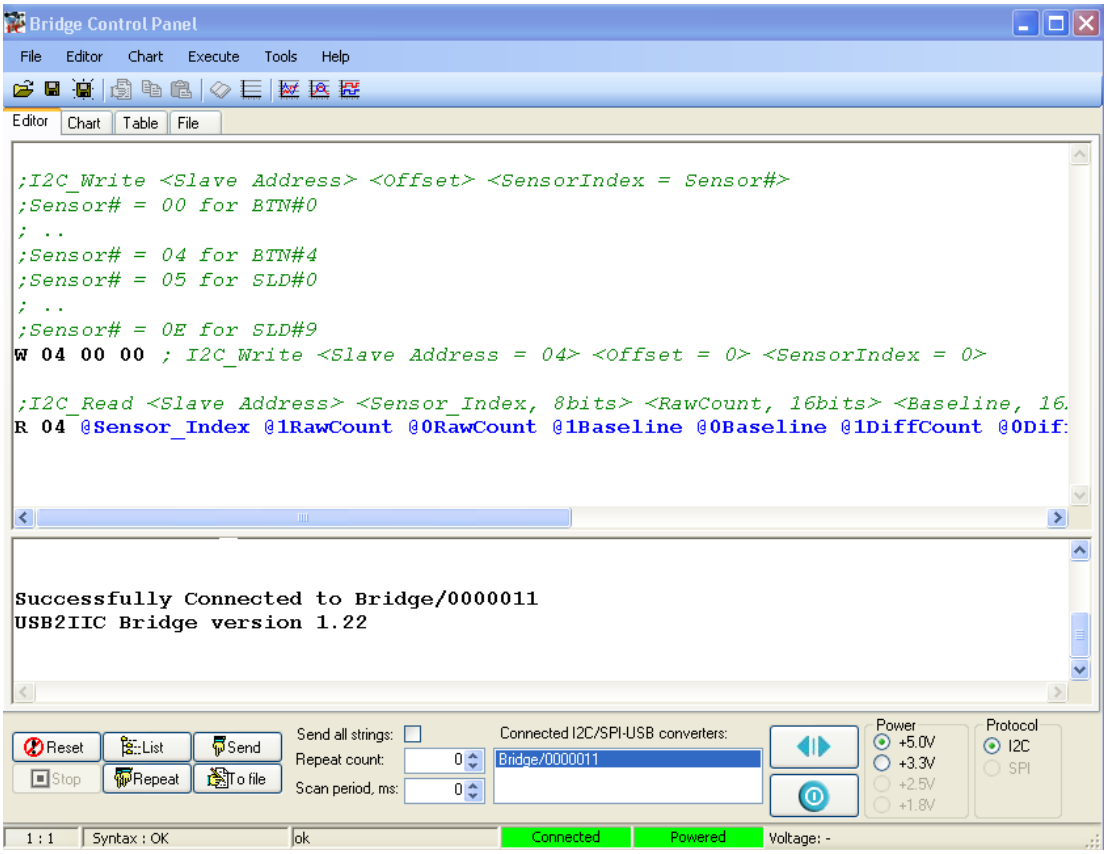
7.8 I²C を経由した CapSense データの読み出し

この手順を使用してサンプルコードを実行し、CapSense データを Bridge Control Panel ツールで読み出します。Bridge Control Panel ツールの詳細については、「AN2397、“CapSense® Data Viewing Tools」を参照してください。

7.8.1 Bridge Control Panel ツールの起動

1. I2USB ブリッジ/ MiniProg3 と USB A－ミニ B ケーブルを使用して、コンピューターを CY3280-20x66 の ISSP コネクタ J3 に接続します。
2. コンピューターのデスクトップで、 **Start** > **All Programs** > **Cypress** > **Bridge Control Panel** (*バージョン番号*) > **Bridge Control Panel** (*バージョン番号*) を選択します。
   Bridge Control Panel は、PSoC Programmer のインストール中にインストールされるコンポーネントです。
3. ポート選択ウィンドウからデバイスを選択します。
4. CY3280-20X66 CapSense コントローラー ボードに 5V で電力を供給します。
5. Bridge Control Panel で **File** > **Open** を選択します。サンプルコード プロジェクト フォルダに含まれる「BCP Configuration Files」フォルダから `BCP.iic` ファイルを読み出します。
6. **Charts** > **Variable Settings** を選択します。サンプルコード プロジェクト フォルダに含まれる「BCP Configuration Files」フォルダから `BCP.ini` ファイルを読み出します。**OK** をクリックして、メイン画面に戻ります。

図 7-3. Bridge Control Panel の表示



7.8.2 BTN0 の raw カウント、ベースライン、差分カウントの読み出し

1. I²C の書き込み命令 W 04 00 00 を一度送信します。
2. **Repeat** ボタンをクリックして、以下の I²C 読み出し命令を継続的に送信します。

   ```
   R 04 @Sensor_Index @1RawCount @0RawCount @1Baseline @0Baseline @1DiffCount @0DiffCount
   @ButtonStatus @SliderPosition
   ```

3. **Chart** タブをクリックして、BTN0 の raw カウント、ベースライン、差分カウントを表示します。

7.8.3 SLD0 の raw カウント、ベースライン、差分カウント、スライダー ポジションの読み出し

1. I²C 書き込み命令 W 04 00 05 を一度送信します。
2. **Repeat** ボタンをクリックして、以下の I²C 読み出し命令を継続的に送信します。

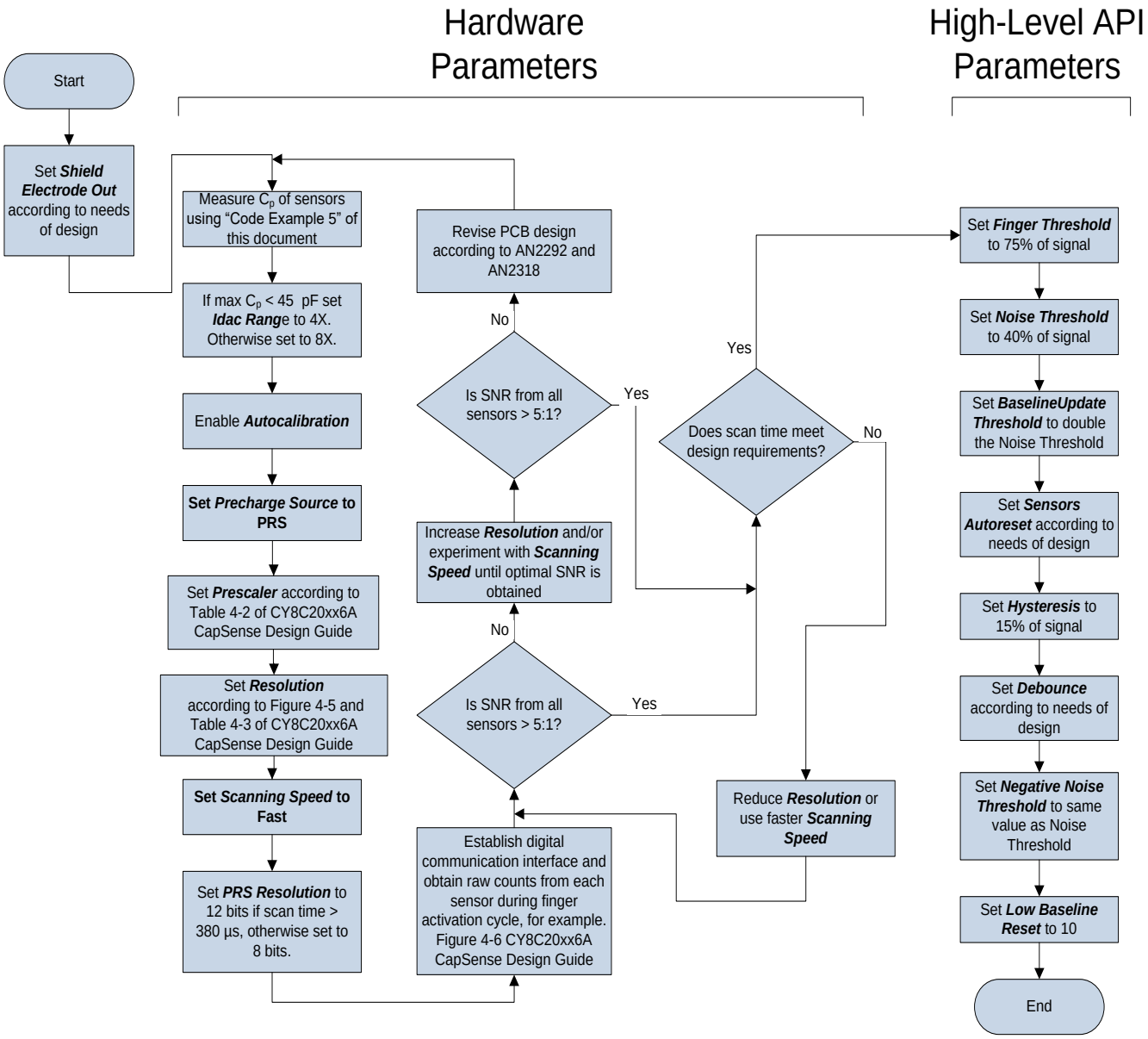
```
R 04 @Sensor_Index @1RawCount @0RawCount @1Baseline @0Baseline @1DiffCount @0DiffCount
@ButtonStatus @SliderPosition
```

3. **Chart** タブをクリックして、SLD0 の raw カウント、ベースライン、差分カウント、スライダー ポジションを表示します。

7.9 CSD ユーザーモジュールの調整

このセクションでは、以下のフローチャートで概説されている手順を用いて、ユーザーモジュール パラメーターを調整します。詳細については、CY8C20xx6A CapSense Design Guide を参照してください。

図 7-4. CSD ユーザーモジュールを調整するためのフローチャート



7.9.1 CSD ユーザーモジュールの設定を行う

PSoC Designer ツールを使用して、CSD ユーザーモジュール パラメーター設定を以下のように入力します。

1. シールド電極は必要ありません。したがって、**ShieldElectrodeOut** パラメーターを **None** に設定します。
2. 表 7-2 では、サンプルコード 3: 絶対センサー静電容量の測定 (CY8C20xx6A CapSense コントローラー使用)で示されるメソッドを使用するセンサーC_P を一覧にします。すべてのセンサーの C_P は 45pF 未満であるため、IDAC 範囲を 4X に設定します。

表 7-2. すべてのセンサーの C_P

センサー番号	ピン	C_P
BTN0	P1[6]	16pF
BTN1	P1[3]	14pF
BTN2	P3[3]	13pF
BTN3	P2[1]	12pF
BTN4	P2[3]	11pF
SLD0	P1[4]	14pF
SLD1	P0[6]	15pF
SLD2	P0[4]	14pF
SLD3	P0[2]	14pF
SLD4	P2[6]	13pF
SLD5	P2[4]	12pF
SLD6	P2[2]	12pF
SLD7	P2[0]	12pF
SLD8	P3[2]	11pF
SLD9	P3[0]	11pF

3. **AutoCalibration** パラメーターを有効にします。
4. **プリチャージ**ソースを **PRS** にセットします。
   PRS ソースはスペクトル拡散により、外部ノイズ源に対するイミュニティを確実にします。
5. すべてのセンサーのうちで最も高い C_P に従って**プリスケーラ**を設定します。この例の最大 C_P は 16pF であるため、プリスケーラーを 4 に設定します。
   詳細については、CSD ユーザーモジュール データシートにある「Prescaler Setting Based on Precharge Source, IMO, and CP」表を参照ください。
6. プロジェクトに異なる C_P を持つ複数のセンサーが含まれる場合、最大の C_P を持つセンサーに従って**分解能**を設定します。
   この例では、最大の C_P は 16pF、Cf は 0.2pF より大きくなります。「CSD ユーザーモジュール データシート」内の「Finger Capacitance（CF） Based on Overlay Thickness and Circular Sensor Diameter」図および「Resolution Setting Based on Finger Capacitance and CP」表を参照して、分解能を 13 に設定してください。
7. **Scan** 速度を **Fast** にセットします。
8. スキャン時間は、380µs より大きくなります。「Scan Time for a Single Sensor in µs Based on Resolution and Scanning Speed」表 (CSD ユーザーモジュール データシートに収録) を参照して、スキャン時間を 720µs、**PRS Resolution** を 8 ビットに設定します。
9. 表 7-3 では、**Bridge Control Panel** で観察される、異なるセンサーのノイズ (指を触れない時の Raw カウントのばらつき) および信号 (指を触れる時の応答) の値を列挙しています。

表 7-3. 全センサーの信号

センサー番号	平均指を触れない時の raw カウント	ノイズ	信号	SNR (整数に丸めた値)
BTN0	7072	24	260	11
BTN1	7017	22	280	13
BTN2	7103	24	280	12
BTN3	7275	22	340	15
BTN4	6953	20	360	18

センサー番号	平均指を触れない時の raw カウント	ノイズ	信号	SNR (整数に丸めた値)
SLD0	6627	26	165	6
SLD1	7071	24	150	6
SLD2	6697	24	150	6
SLD3	6505	24	145	6
SLD4	6028	22	160	7
SLD5	5840	28	155	5
SLD6	5687	26	160	6
SLD7	5598	24	150	6
SLD8	5362	24	150	6
SLD9	5110	30	160	5

10. 全センサーの SNR が 5:1 を超え、スキャン時間は指で押すと **LED が瞬間的に光る程度に短いため、表 7-4 で説明しているようにしきい値のパラメーターを設定することができます。**
    閾値を設定するには、全センサーの中で最も低い信号の値を基準として使用します。したがって、この例では、信号の値は 145 です。

    表 7-4. CSD UM の高級パラメーター

閾値パラメーター	推奨値	値	UM パラメーター設定
指の閾値	信号の 75%	108.75	108
ノイズ閾値	信号の 40%	58	58
ベースライン更新閾値	2*ノイズ閾値	116	116
マイナスのノイズ閾値	ノイズ閾値	58	58
ヒステリシス	信号の 15%	21.75	21

11. スキャン時間の要件を満たしており、誤検出がレポートされていないので、**Low Baseline Reset** パラメーターを 10 にセットします。**Debounce** を 3 にセットします。

サンプルコード 8　フィードバック抵抗付き CSD の調整 (CY8C21x34/B CapSense コントローラー使用)

8.1　プロジェクト名

CE_8_TuningCSD_withCY8C21x34/B

8.2　概要

このサンプルコードでは、CY8C21x34/B デバイスの CSD ユーザーモジュール (UM) を調整する方法を示します。CapSense ユーザーモジュールをチューニングするには、CapSense データをグラフ形式で表示できなければなりません。このサンプルコードでは、EzI2C UM を使用して必要な CapSense パラメーターを I^2C バスと I2USB ブリッジに転送します。PC で Bridge Control Panel を使用して、このデータを表示します。

8.3　ハードウェアのセットアップ

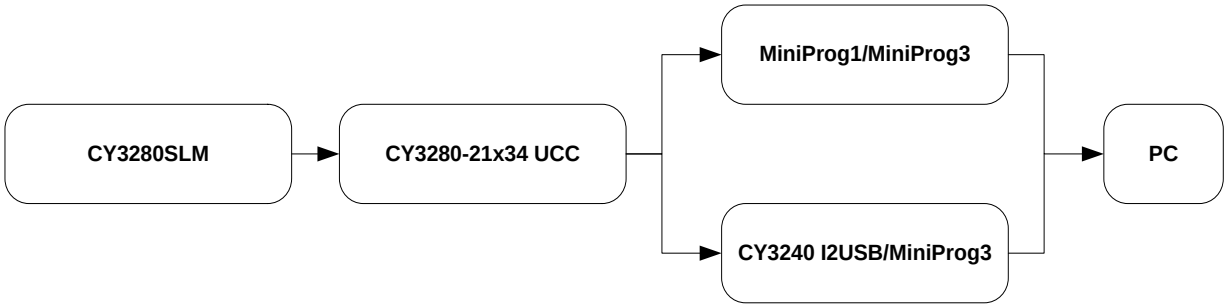
8.3.1　要求事項

- CY3280-21x34 ユニバーサル CapSense コントローラー ボード
- CY3280-SLM ユニバーサル CapSense リニア スライダー モジュール
- CY3240-I2USB ブリッジまたは CY8CKIT-002 Miniprog3
- CY3217-MiniProg1 プログラマ キットまたは CY8CKIT-002 Miniprog3
- USB A－ミニ B ケーブル
- Windows XP 以降の OS を実行する PC

8.3.2　アセンブリ

図 8-1 は、ハードウェアのセットアップを示します。CY3280-21x34 UCC キットを CY3280-SLM モジュールに 22x2_RA_Receptacle 経由で接続します。MiniProg1/Miniprog3 または I2USB ブリッジのどちらかを、キットの ISSP ヘッダーに接続します。ハードウェア準備では、プログラミングに MiniProg1/Miniprog3 を使用し、PC にデータを送信するには I2USB ブリッジ/Miniprog3 を使用します。 これらは、USB ケーブルで PC に接続します。

図 8-1. ハードウェア セットアップのブロック図

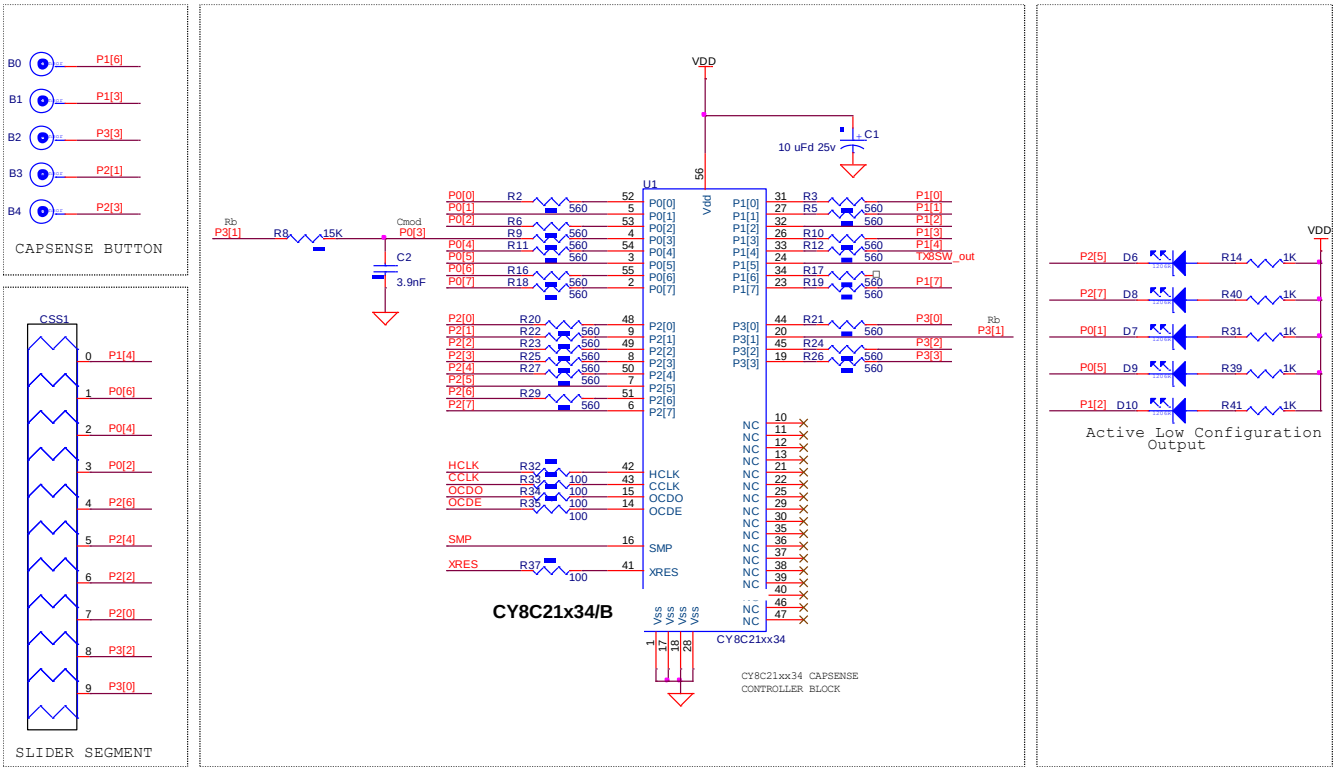


8.3.3 ボードの設定

以下のハードウェアを接続します。

- CY3280-SLM ドーターカードのヘッダーJ1 を CY3280-21x34 UCC ボード上の 22x2_RA_Receptacle (コネクタ P2) に接続します。
- ヘッダーJ7 にジャンパーを嵌め、UCC ボードの 5V ピンと VCC_PROG のピンを接続させます。この設定により、ISSP コネクタから CapSense コントローラーに電力を供給できます。
- ヘッダーJ4 にジャンパーを嵌め、UCC ボードの XRES ピンと XRES/INT ピン(ピン 1 と 2) を接続させます。この設定では、CapSense コントローラーの XRES ピンが ISSP コネクタ J3 のピン 3 に接続されます。
- ヘッダーJ2 にジャンパーを嵌め、CY3280-SLM ボードの GND ピンと SHILED ピン (ピン 2 と 3) を接続させます。この設定により、CY3280-SLM ボード上のハッチ パターンが接地されます。
- MiniProg1/MiniProg3 を UCC ボードの J3 上の ISSP ヘッダーに接続します。この接続は、コードで生成される hex ファイルで UCC をプログラミングするときのみ要求されます。この接続は、CapSense データが Bridge Control Panel ソフトウェアで読み出される時、I2USB ブリッジ/Miniprog3 と置き換える必要があります。
- USB Aーミニ B ケーブルを使用して、MiniProg1 (または I2USB ブリッジ、あるいは MiniProg3) の他方の端に接続します。

8.4 回路図



変調コンデンサ (C_{MOD}) は、P0[3] に接続された 2.2nF のコンデンサです。560Ω の抵抗は、RF 干渉を軽減するため、各 CapSense ボタンに直列に接続されます。LED は 1kΩ の直列抵抗に接続しアクティブローの設定です。全部で、LED が 5 個、ボタンが 5 個、セグメントが 10 あるスライダーが 1 個あります。

表 8-1. LED、ボタン、スライダー セグメントのピン割り当て

LED	ボタン	スライダー セグメント
LED0 - P2[5]	BTN0 - P1[6]	SLD0 - P1[4]、SLD1 - P0[6]
LED1 - P2[7]	BTN1 - P1[3]	SLD2 - P0[4]、SLD3 - P0[2]
LED2 - P0[1]	BTN2 - P3[3]	SLD4 - P2[6]、SLD5 - P2[4]
LED3 - P0[5]	BTN3 - P2[1]	SLD6 - P2[2]、SLD7 - P2[0]
LED4 - P1[2]	BTN4 - P2[3]	SLD8 - P3[2]、SLD9 - P3[0]

8.5 ソフトウェアの準備

8.5.1 ツール

- PSoC® Designer (バージョン 5.2 以降)
- PSoC Programmer (バージョン 3.13 以降)
- Bridge Control Panel (ブリッジコントロールパネル)

8.5.2 ユーザーモジュールと配置

以下の表は、このコード表で使用されているユーザーモジュール 、および各ユーザーモジュールが占有するハードウェア リソースを一覧にします。

ユーザーモジュール	配置
CSD	CapSense ブロック、タイマー1 (デフォルト)
EzI2C	I²C/SPI ブロック

8.5.3 ユーザーモジュール パラメーター、グローバル リソース

プロジェクトの `ReadMe.txt` ファイルでは、このサンプルコードで使用されるユーザーモジュールのパラメーター設定について説明しています。また、グローバルリソースのリストが含まれています。

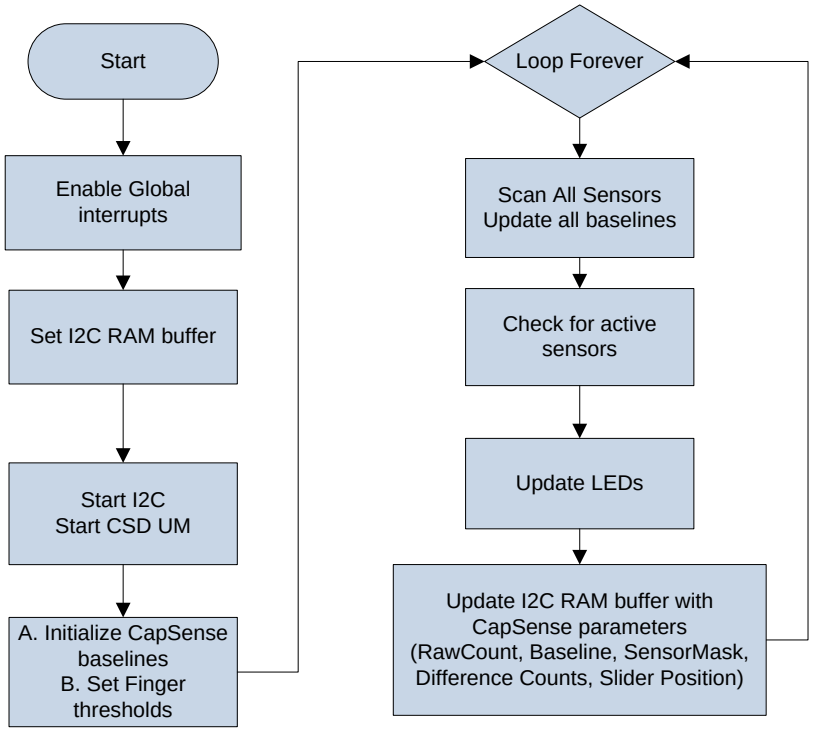
注: サンプルコードは、 CY3280-SLM キットに付属するアクリル製オーバーレイ (1.5mm) 向けに調整しました。オーバーレイがこれより厚い場合は、このサンプルコードで説明しているプロジェクトを調整します。

8.6 動作

リセットすると、ファームウェアは以下の動作を実行します。

- 構造体 MyI2C_Regs を定義し、ボタン番号、raw カウント、差分カウント、ベースライン、セントロイド位置、所定のボタン番号に対応する CapSense ボタンのステータスを保管します。
- グローバル割り込みを有効にしてから、CSD ユーザーモジュールを開始します。
- EzI2C UM を開始し、構造体 MyI2C_Regs を I²C RAM バッファとして設定します。
- 無限ループで以下の操作を実行します。
  - すべてのセンサーを継続的にスキャンし、構造体 MyI2C_Regs を raw カウント、差分カウント、ベースライン、スライダのセントロイド位置、および要求される CapSense ボタンで更新します。I²C マスターは、ボタン番号を EzI2C スレーブの I²C バッファの最初のバイトに書き込み、特定のボタンの CapSense を要求することができます。
  - ファームウェアはボタンが押されるのを検出すると、対応する LED を点灯させます。ボタンを放すと、LED を消灯させます。
  - スライダーに触れると、ファームウェアは LED を点灯させて接触位置を示します。

図 8-2. サンプルコード 8 の機能フローチャート



8.7 コードの実行

ボードにサンプル プロジェクトをプログラムしてから、この手順を使用してサンプルコードを実行します。UCC ボードのプログラミングの方法については、CY3280-21x34 キットガイドの第 5 章を参照してください。

1. ボードを 5V で給電するには、MiniProg1/Miniprog3 または CY3280-21x34 UCC キットガイドで説明している電源のいずれかを使用します。
2. CY3280-SLM モジュール ボードでリニア スライダーに触れます。
   CY3280-SLM モジュール ボードの対応する LED が点灯します。
3. ボタンに触れます。
   CY3280-SLM モジュール ボードの対応する LED が点灯します。同時に複数のボタンが機能します。リニア スライダーとボタンも同時に使用できます。

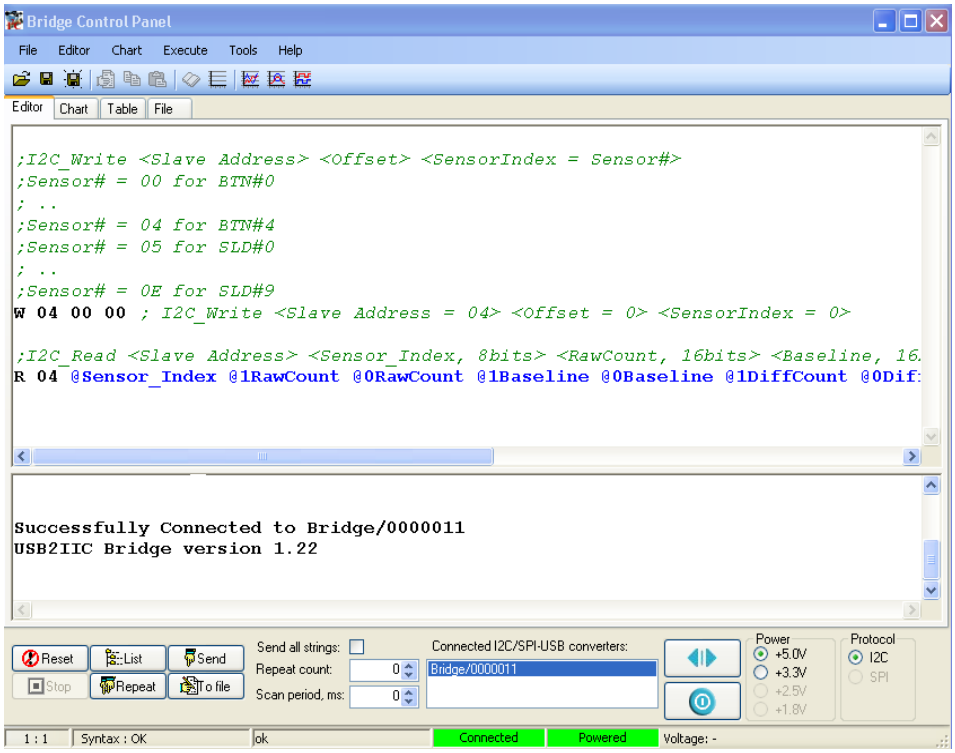
8.8 I²C を経由した CapSense データの読み出し

この手順を使用してサンプルコードを実行し、CapSense データを Bridge Control Panel ツールで読み出します。Bridge Control Panel ツールの詳細については、「AN2397、“CapSense® Data Viewing Tools」を参照してください。

8.8.1 Bridge Control Panel ツールの起動

1. I2USB ブリッジ/MiniProg3 と USB A－ミニ B ケーブルを使用して、コンピューターを CY3280-21x34 ユニバーサル CapSense コントロール ボードの ISSP コネクタ J3 に接続します。
2. コンピューターのデスクトップで、**Start** > **All Programs** > **Cypress** > **Bridge Control Panel** *(バージョン番号)* > **Bridge Control Panel** *(バージョン番号)* を選択します。
   Bridge Control Panel は、PSoC Programmer のインストール中にインストールされるコンポーネントです。
3. ポート選択ウィンドウからデバイスを選択します。
4. CY3280-21X34 CapSense コントローラー ボードに 5V で電力を供給します。

5. Bridge Control Panel で **File** > **Open** を選択します。サンプルコード プロジェクト フォルダに含まれる「BCP Configuration Files」フォルダから `BCP.iic` ファイルを読み出します。
6. **Charts** > **Variable Settings** を選択します。サンプルコード プロジェクト フォルダに含まれる「BCP Configuration Files」のフォルダから `BCP.ini` ファイルを読み出します。**OK** をクリックして、メイン画面に戻ります。



8.8.2 BTN0 の raw カウント、ベースライン、差分カウントの読み出し

1. I²C の書き込み命令 W 04 00 00 を一度送信します。
2. **Repeat** ボタンをクリックして、以下の I²C 読み出し命令を継続的に送信します。

```
R 04 @Sensor_Index @1RawCount @0RawCount @1Baseline @0Baseline @1DiffCount @0DiffCount
@ButtonStatus @SliderPosition
```

3. **Chart** タブをクリックして、BTN0 の raw カウント、ベースライン、差分カウントを表示します。

8.8.3 SLD0 の raw カウント、ベースライン、差分カウント、スライダー ポジションの読み出し

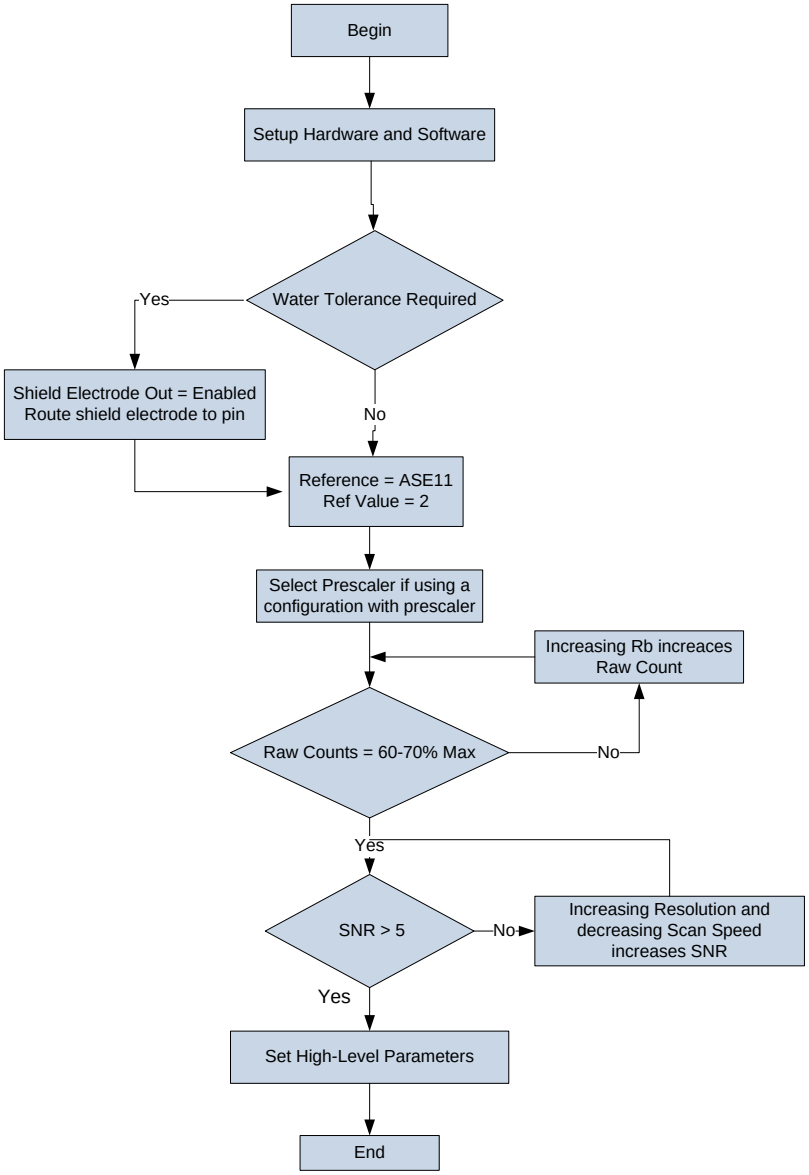
1. I²C 書き込み命令 W 04 00 05 を一度送信します。
2. **Repeat** ボタンをクリックして、以下の I²C 読み出し命令を継続的に送信します。

```
R 04 @Sensor_Index @1RawCount @0RawCount @1Baseline @0Baseline @1DiffCount @0DiffCount
@ButtonStatus @SliderPosition
```

3. **Chart** タブをクリックして、SLD0 の raw カウント、ベースライン、差分カウント、スライダ ポジションを表示します。

8.9 CSD ユーザーモジュールの調整

このセクションでは、以下のフローチャートで概説されている手順を用いて、ユーザーモジュールパラメーターを調整します。詳細については、CY8C21x34/B CapSense design guide を参照してください。



8.9.1 CSD UM の設定を行う

PSoC Designer ツールを使用して、CSD UM パラメーター設定を以下のように入力します。

1. このサンプルコードは、Rb = 15kΩ および C_{MOD} = 10nF を使用します。CY3280-21x34 UCC ボードは、固定抵抗 Rb を伴います。カスタムボード デザインについては、Rb を調整してユーザーモジュールを調整します。
   CapSense センサーの適切な感度を得るために Rb の最適値を選択する方法については、CY8C21x34/B CapSense design Guide のセクション 4.2.3 を参照してください。
2. シールド電極は必要ありません。したがって、**ShieldElectrodeOut** パラメーターを **None** に設定します。

表 8-2 では、サンプルコード 5: 絶対センサー静電容量の測定 (CY8C21x34/B CapSense コントローラー使用) で示されるメソッドを使用して測定した各センサーの C_P を一覧にします。この例の最大 CP は 16pF であるため、プリスケーラを 4 に設定します。

表 8-2. 全てのセンサーの C_P

センサー	ピン	C_P
BTN0	P1[6]	16pF
BTN1	P1[3]	14pF
BTN2	P3[3]	13pF
BTN3	P2[1]	12pF
BTN4	P2[3]	11pF
SLD0	P1[4]	14pF
SLD1	P0[6]	15pF
SLD2	P0[4]	14pF
SLD3	P0[2]	14pF
SLD4	P2[6]	13pF
SLD5	P2[4]	12pF
SLD6	P2[2]	12pF
SLD7	P2[0]	12pF
SLD8	P3[2]	11pF
SLD9	P3[0]	11pF

3. **プリチャージソース**を **PRS** に設定します。
   PRS ソースはスペクトル拡散により、外部ノイズ源に対するイミュニティを確実にします。
4. すべてのセンサーのうちで最も高い C_P に従って**プリスケーラ**を設定します。この例の最大 C_P は 16pF であるため、プリスケーラを 4 に設定します。
   詳細については、CSD UM データシートにある「Prescaler Setting Based on Precharge Source, IMO, and CP」表を参照ください。
5. プロジェクトに異なる C_P を持つ複数のセンサーが含まれる場合、最大の C_P を持つセンサーに従って**分解能**を設定します。この例では、最大の C_P は 16pF、Cf は 0.2pF より大きくなります。「CSD UM データシート」での「Finger Capacitance (CF) Based on Overlay Thickness and Circular Sensor Diameter」図および「Resolution Setting Based on Finger Capacitance and CP」表を参照して、分解能を 13 に設定してください。
6. **Comparator** リファレンスを ASE11 に設定します。
   ASE11 は、大部分のアプリケーションに推奨されます。
7. **Scan** 速度を **Normal** に設定します。

表 8-3 では、**Bridge Control Panel** で観察される、異なるセンサーのノイズ (指を触れない時の Raw カウントのばらつき) および信号 (指を触れる時の応答) の値を列挙しています。

表 8-3. 全センサーの信号

センサー	指を触れない時の平均 raw カウント	信号	ノイズ	SNR (整数に丸めた値)
BTN0	4451	179	20	9
BTN1	3921	182	20	9
BTN2	3419	184	21	9
BTN3	3186	179	20	9
BTN4	2652	191	21	9
SLD0	3790	100	20	5
SLD1	4432	101	20	5
SLD2	4026	101	19	5
SLD3	4011	103	19	5
SLD4	3739	107	20	5
SLD5	3456	104	20	5
SLD6	3310	110	21	5
SLD7	3158	99	19	5
SLD8	2900	101	19	5
SLD9	2688	115	20	5

8. 全センサーの SNR は 5:1 を超え、スキャン時間は指で押すと LED が同時に光るほど低いため、表 8-4 で説明しているようにしきい値のパラメーターを設定することができます。
   閾値を設定するには、全センサーの中で最も低い信号の値を基準として使用します。したがって、この例では、信号の値は 99 です。

   表 8-4. CSD UM の高級パラメーター

閾値パラメーター	推奨値	値	UM パラメーター設定
指の閾値	信号の 75%	75.3	74
ノイズ閾値	信号の 40%	39.6	40
ベースライン更新閾値	2 ノイズ閾値	80	80
マイナスのノイズ閾値	ノイズ閾値	40	40
ヒステリシス	信号の 15%	14.9	15

9. スキャン時間の要件を満たしており、誤検出がレポートされていないので、**Low Baseline Reset** パラメーターを 10 にセットします。**Debounce** を 3 にセットします。

サンプルコード 9　CSA_EMC の調整 (CY8C20x34 CapSense コントローラー使用)

9.1　プロジェクト名

CE_9_TuningCSAEMC_withCY8C20x34

9.2　概要

このサンプルコードでは、CY8C20x34 デバイスの CSA_EMC ユーザーモジュール (UM）) を調整する方法を示します。CapSense UM を調整するには、CapSense データをグラフ形式で表示できなければなりません。このサンプルコードでは、EzI2C UM を使用して必要な CapSense パラメーターを I^2C バスと I^2C-USB ブリッジを経由して転送します。PC でブリッジコントロールパネルを使用して、このデータを表示します。

9.3　ハードウェアの準備

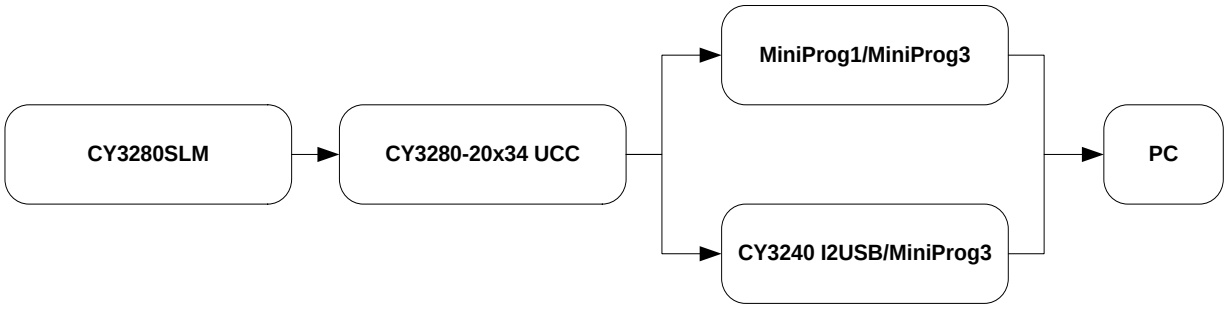
9.3.1　関連ハードウェア

- CY3280-20x34 ユニバーサル CapSense コントローラー ボード
- CY3280-SLM ユニバーサル CapSense リニア スライダー モジュール
- CY3240-I2USB ブリッジまたは CY8CKIT-002 Miniprog3
- CY3217-MiniProg1 プログラマ キットまたは CY8CKIT-002 Miniprog3
- USB Aーミニ B ケーブル
- USB ポート搭載 PC

9.3.2　アセンブリ

図 9-1 は、ハードウェアのセットアップを示します。CY3280-20x34 UCC キットを CY3280-SLM モジュールに 22x2_RA_Receptacle 経由で接続します。MiniProg1/Miniprog3 または I2USB ブリッジのどちらかを、キットの ISSP ヘッダーに接続します。ハードウェア準備では、プログラミングに MiniProg1/Miniprog3 を使用し、PC にデータを送信するには I2USB ブリッジまたは Miniprog3 を使用します。これらは、USB ケーブルで PC に接続します。

図 9-1. ハードウェア セットアップのブロック図

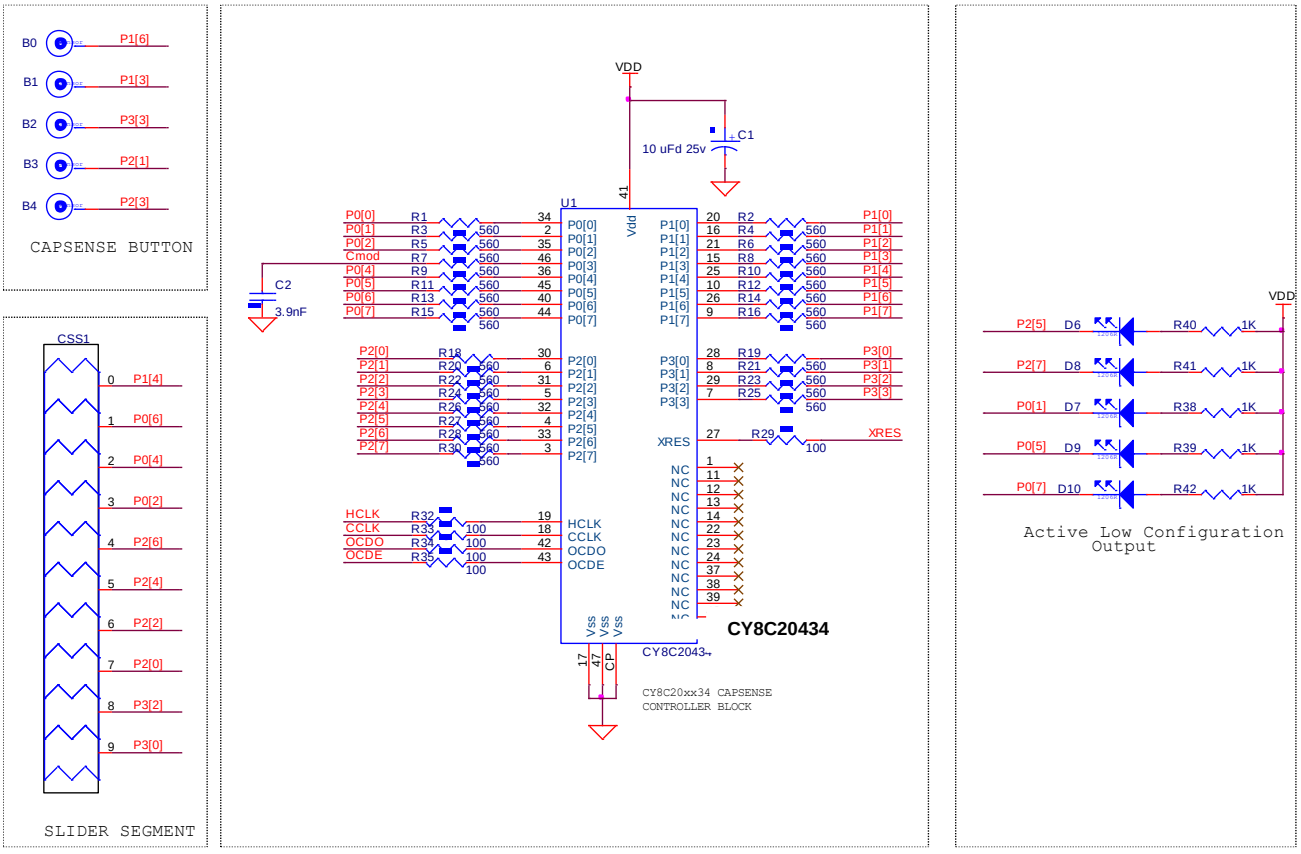


9.3.3 ボードの設定

以下のハードウェアを接続します。

- CY3280-SLM ドーターカードのヘッダーJ1 を CY3280-20x34UCC ボード上の 22x2_RA_Receptacle (コネクタ P2) に接続します。
- ヘッダーJ1 にジャンパーを嵌め、UCC ボードの V_{CC} ピンと 5 V をショートさせます。この設定により、ISSP コネクタから CapSense コントローラーに電力を供給できます。
- ヘッダーJ4 にジャンパーを嵌め、UCC ボードの XRES ピンと XRES/INT ピン(ピン 1 と 2) を接続させます。この設定では、CapSense コントローラーの XRES ピンが ISSP コネクタ J3 のピン 3 に接続されます。
- ヘッダーJ2 にジャンパーを嵌め、CY3280-SLM ボードの GND ピンと SHILED ピン (ピン 2 と 3) を接続させます。この設定により、CY3280-SLM ボード上のハッチパターンが接地されます。
- MiniProg1/MiniProg3 を UCC ボードの J3 上の ISSP ヘッダーに接続します。この接続は、コードで生成される hex ファイルで UCC をプログラミングするときのみ要求されます。この接続は、CapSense データが Bridge Control Panel ソフトウェアで読み出される時、I2USB ブリッジ/MiniProg3 と置き換える必要があります。
- USB A－ミニ B ケーブルを使用して、MiniProg1/Miniprog3 (または I2USB ブリッジ) の他の端を接続します。

9.4 回路図



変調コンデンサ (C_{MOD}) は、P0[3] に接続された 2.2nF のコンデンサです。560Ω の抵抗は、RF 干渉を軽減するため、各 CapSense ボタンに直列に接続されます。LED は 1kΩ の直列抵抗に接続しアクティブローの設定です。全部で、LED が 5 個、ボタンが 5 個、セグメントが 10 あるスライダが 1 個あります。

表 9-1. LED、ボタン、スライダー セグメントのピン割り当て

LED	ボタン	スライダー セグメント
LED0 - P2[5]	BTN0 - P1[6]	SLD0 - P1[4]、SLD1 - P0[6]
LED1 - P2[7]	BTN1 - P1[3]	SLD2 - P0[4]、SLD3 - P0[2]
LED2 - P0[1]	BTN2 - P3[3]	SLD4 - P2[6]、SLD5 - P2[4]
LED3 - P0[5]	BTN3 - P2[1]	SLD6 - P2[2]、SLD7 - P2[0]
LED4 - P1[2]	BTN4 - P2[3]	SLD8 - P3[2]、SLD9 - P3[0]

9.5 ソフトウェアの準備

9.5.1 ツール

- PSoC® Designer (バージョン 5.2 以降)
- PSoC Programmer (バージョン 3.13 以降)
- Bridge Control Panel (ブリッジ コントロール パネル)

9.5.2 ユーザーモジュール

以下の表は、このコード表で使用されているユーザーモジュール 、および各ユーザーモジュールが占有するハードウェア リソースを一覧にします。

ユーザーモジュール	ハードウェア リソース
CSA_EMC	CapSense ブロック
EzI2C	I²C ブロック

9.5.3 ユーザーモジュール パラメーター、グローバルリソース

プロジェクトの `ReadMe.txt` ファイルでは、このサンプルコードで使用されるユーザーモジュールのパラメーター設定について説明しています。また、グローバルリソースのリストが含まれています。

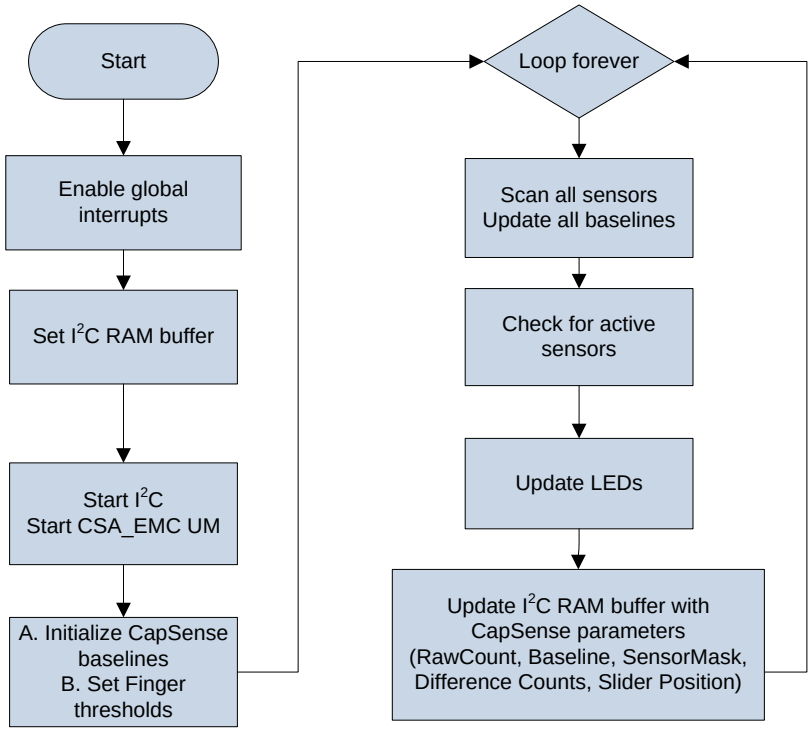
注: サンプルコードは、CY3280-SLM キットに付属するアクリル製オーバーレイ (1.5mm) 向けに調整しました。オーバーレイがこれより厚い場合は、このサンプルコードで説明しているプロジェクトを調整します。

9.6 動作

リセットすると、ファームウェアは以下の操作を実行します。

- 構造体 MyI2C_Regs を定義し、ボタン番号、Raw カウント、差分カウント、ベースライン、セントロイド位置、所定のボタン番号に対応する CapSense ボタンのステータスを保管します。
- グローバル割り込みを有効にしてから、CSA_EMC UM を開始します。
- EzI2C のユーザーモジュールを開始し、構造体 MyI2C_Regs を I²C RAM バッファとして設定します。
- 無限ループで以下の操作を実行します。
  - すべてのセンサーを継続的にスキャンし、構造体 MyI2C_Regs を Raw カウント、差分カウント、ベースライン、スライダのセントロイド位置、および要求される CapSense ボタンで更新します。I²C マスターは、ボタン番号を EzI2C スレーブの I²C バッファの最初のバイトに書き込み、特定のボタンの CapSense を要求することができます。
  - ファームウェアはボタンが押されるのを検出すると、対応する LED を点灯させます。ボタンを放すと、LED を消灯させます。
  - スライダーに触れると、ファームウェアは LED を点灯させて接触位置を示します。

図 9-2. サンプルコード 9 の機能フローチャート



9.7 コードの実行

ボードにサンプルプロジェクトをプログラムしてから、この手順を使用してサンプルコードを実行します。UCC ボードのプログラミングの方法については、CY3280-20x34 キットガイドの第 5 章を参照してください。

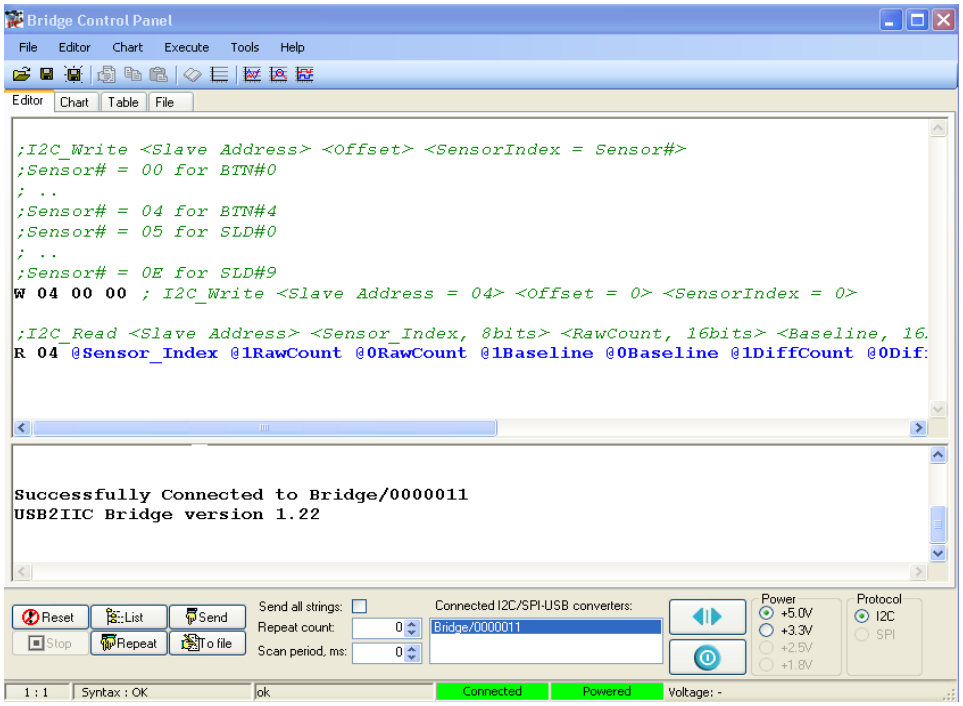
1. ボードを 5V で給電するには、MiniProg1/Miniprog3 または CY3280-20x34 UCC キットガイドで説明している電源のいずれかを使用します。
2. CY3280-SLM モジュール ボードでリニア スライダーに触れます。
   CY3280-SLM モジュール ボードの対応する LED が点灯します。
3. ボタンに触れます。
   CY3280-SLM モジュール ボードの対応する LED が点灯します。同時に複数のボタンが機能します。リニア スライダーとボタンも同時に使用できます。

9.8 I²C を経由した CapSense データの読み出し

この手順を使用してサンプルコードを実行し、CapSense データを Bridge Control Panel ツールで読み出します。Bridge Control Panel ツールの詳細については、「AN2397、“CapSense® Data Viewing Tools」を参照してください。

1. I2USB ブリッジ/MiniProg3 と USB A－ミニ B ケーブルを使用して、コンピューターを CY3280-20x34 ユニバーサル CapSense コントロール ボードの ISSP コネクタ J3 に接続します。
2. コンピューターのデスクトップで、**Start** > **All Programs** > **Cypress** > **Bridge Control Panel** *(バージョン番号)* > **Bridge Control Panel** *(バージョン番号)* を選択します。
   Bridge Control Panel は、PSoC Programmer のインストール中にインストールされるコンポーネントです。
3. ポート選択ウィンドウからデバイスを選択します。
4. CY3280-20X34 CapSense コントローラー ボードに 5V で電力を供給します。
5. Bridge Control Panel で **File** > **Open** を選択します。サンプルコード プロジェクト フォルダに含まれる「BCP Configuration Files」フォルダから BCP.iic ファイルを読み出します。

6. **Charts** > **Variable Settings** を選択します。サンプルコード プロジェクト フォルダに含まれる「BCP Configuration Files」フォルダから `BCP.ini` ファイルを読み出します。**OK** をクリックして、メイン画面に戻ります。



9.8.1 BTN0 のローカウント、ベースライン、差分カウントの読み出し

1. I²C の書き込み命令 W 04 00 00 を一度送信します。
2. **Repeat** ボタンをクリックして、以下の I²C 読み出し命令を継続的に送信します。

```
R 04 @Sensor_Index @1RawCount @0RawCount @1Baseline @0Baseline @1DiffCount @0DiffCount
@ButtonStatus @SliderPosition
```

3. **Chart** タブをクリックして、BTN0 の raw カウント、ベースライン、差分カウントを表示します。

9.8.2 SLD0 の raw カウント、ベースライン、差分カウント、スライダー ポジションの読み出し

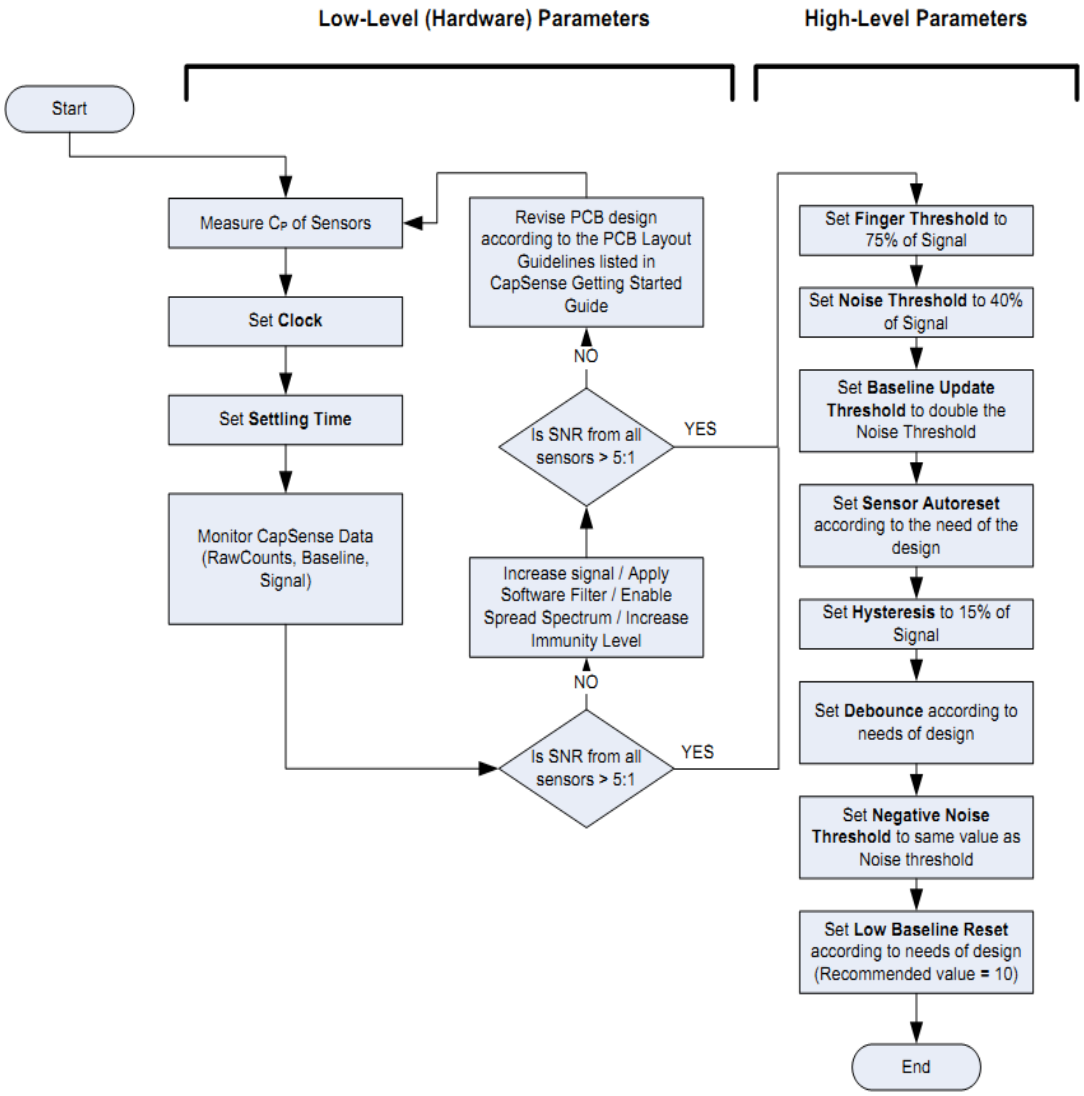
1. I²C 書き込み命令 W 04 00 05 を一度送信します。
2. **Repeat** ボタンをクリックして、以下の I²C 読み出し命令を継続的に送信します。

```
R 04 @Sensor_Index @1RawCount @0RawCount @1Baseline @0Baseline @1DiffCount @0DiffCount
@ButtonStatus @SliderPosition
```

3. **Chart** タブをクリックして、SLD0 の Raw カウント、ベースライン、差分カウント、スライダー ポジションを表示します。

9.9 CSA_EMC ユーザーモジュールの調整

このセクションでは、以下のフローチャートで概説されている手順を用いて、ユーザーモジュール パラメーターを調整します。詳細については、CY8C20x34 CapSense design guide を参照してください。



9.9.1 CSD ユーザーモジュールの設定を行う

PSoC Designer ツールを使用して、CSD UM パラメーター設定を以下のように入力します。

1. CPU_CLK を PSoC Designer ツールのグローバル リソース ウィンドウにある SysClk/2 と等しく設定します。設計ニーズに合わせて、グローバル リソース ウィンドウの残りのパラメーターを設定します。
2. 表 9-2 では、サンプルコード 6: 絶対センサー静電容量の測定 (CY8C20x34 CapSense コントローラー使用)で示されるメソッドを使用して測定した各センサーの C_P を一覧にします。この例の最大 C_P は 15pF で、IMO は 12MHz であるため、クロックを IMO/4 に設定します。
   IMO の詳細については、CY8C20x34 CapSense Design Guide の表 15 を参照してください。

表 9-2. すべてのセンサーの C_P

センサー	ピン	C_P
BTN0	P1[6]	15pF
BTN1	P1[3]	13pF
BTN2	P3[3]	12pF
BTN3	P2[1]	11pF
BTN4	P2[3]	9pF
SLD0	P1[4]	14pF
SLD1	P0[6]	14pF
SLD2	P0[4]	13pF
SLD3	P0[2]	12pF
SLD4	P2[6]	12pF
SLD5	P2[4]	11pF
SLD6	P2[2]	10pF
SLD7	P2[0]	10pF
SLD8	P3[2]	10pF
SLD9	P3[0]	9pF

3. 整定時間のパラメーターを式を用いて計算します。

$$\text{Settling Time} = \frac{5 \times C_{INT}}{\left(\text{Clock} \times C_P \times 25 \left(\frac{1}{F_{CPU}}\right)\right)}$$

C_{INT} = 3.9nF

クロック = IMO/4 = 3MHz

C_P = 15pF

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

整定時間の上記の式で値を置き換えると、「Settling Time = 80」

新しい設定を実行する場合、Bridge Control Panel ツールを用いて SNR を測定します。表 9-3 では、このサンプルコードで見られる値を列挙します。

表 9-3. 全センサーで観測される信号

センサー番号	平均指を触れない時の Raw カウント	ノイズ	信号	SNR (整数に丸めた値)
BTN0	2000	14	130	9
BTN1	2140	15	140	9
BTN2	2210	18	140	8
BTN3	2230	16	140	9
BTN4	1930	20	150	7
SLD0	1985	11	100	9
SLD1	2310	14	95	7
SLD2	2075	14	105	7

センサー番号	平均指を触れない時の Raw カウント	ノイズ	信号	SNR (整数に丸めた値)
SLD3	2160	13	95	7
SLD4	2500	14	100	7
SLD5	2370	15	105	7
SLD6	2200	16	105	7
SLD7	2180	14	110	8
SLD8	1960	14	110	8
SLD9	2050	15	110	7

4. 最低 SNR は 7 (整数に四捨五入) で、5:1 より大きいため、高級パラメーターを表 9-4 で説明されているように設定します。

表 9-4. UM の高級パラメーター

閾値	推奨値	値	ユーザーモジュール パラメーター設定
指の閾値	信号の 75%	71.25	71
ノイズ閾値	信号の 40%	38	38
ベースライン更新閾値	2* ノイズ閾値	76	76
マイナスのノイズ閾値	ノイズ閾値	38	38
ヒステリシス	信号の 15%	14.25	14

5. 他のパラメーターは初期値のままにします。

サンプルコード 10　CY8C20xx6A の SmartSense による CapSense ボタンの消費電力の最適化

10.1 プロジェクト名

CE_10_PowerOptimization_withCY8C20xx6A

10.2 概要

このサンプルコードの目的は、SmartSense を使用するひとつのボタンの消費電流を 50µA 未満にすることにあります。このサンプルコードは、SmartSense と 16 ビット タイマーのユーザーモジュールを組み込んでいます。このデバイスは、一定期間スリープモードで動作して、平均消費電流を減らします。16 ビット タイマーは、スリープ時間経過後にデバイスを復帰させるために、スキャン タイマーとして使用します。

10.3 ハードウェアの準備

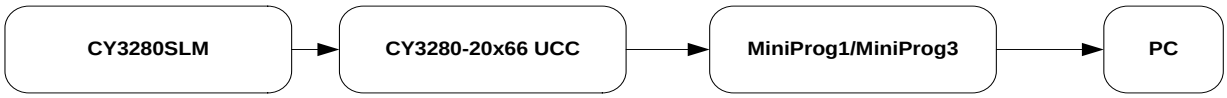
10.3.1 関連ハードウェア

- CY3280-20x66 ユニバーサル CapSense コントローラー ボード
- CY3280-BSM ユニバーサル CapSense シンプル ボタン モジュール
- CY3217-MiniProg1 プログラマ キットまたは CY8CKIT-002 Miniprog3
- USB A－ミニ B ケーブル
- 電流計
- 電源

10.3.2 組み立て

図 10-1 はハードウェアの準備を示します。CY3280-20x66 UCC ユニットを CY3280-BSM モジュールに 22x2_RA_Receptacle 経由で接続します。この構成では、MiniProg1/MiniProg3 を使用してキットの ISSP ヘッダーに接続してプログラミングします。

図 10-1. ハードウェア セットアップのブロック図

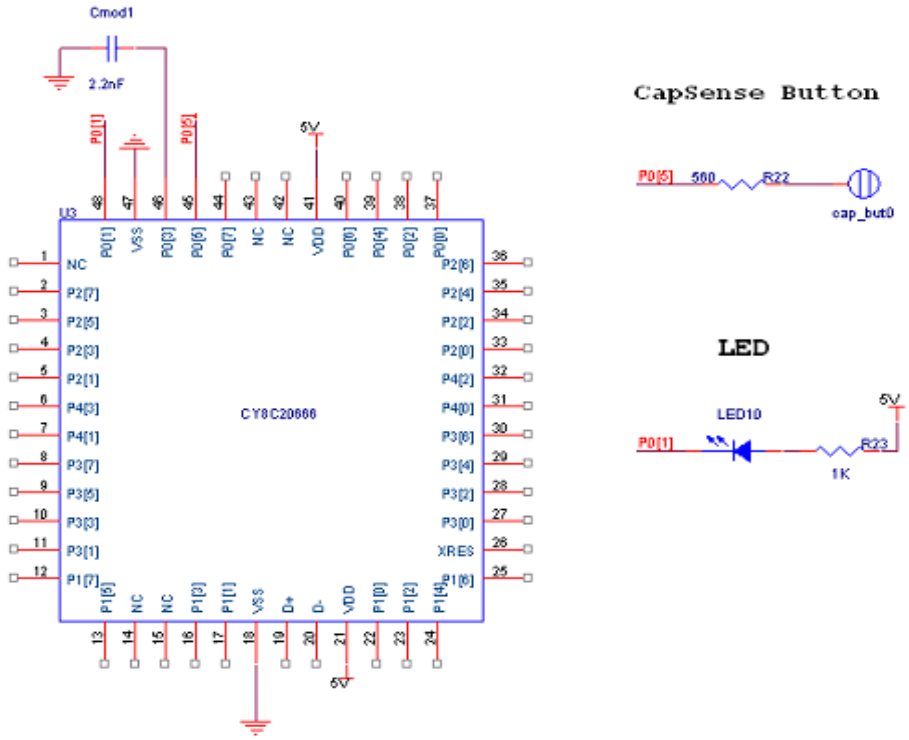


10.4 ボードの設定

以下のハードウェアを接続します。

- CY3280-BSM ドーターカードのヘッダーJ1 を CY3280-20x66 UCC ボード上の 22x2_RA_Receptacle (コネクタ P2) に接続します。
- ヘッダーJ7 にジャンパーを嵌め、UCC ボードの V_{CC} ピンと V_{CC_PROG} ピンを接続させます。この設定により、ISSP コネクタから CapSense コントローラーに電力を供給できます。
- ヘッダーJ4 にジャンパーを嵌め、UCC ボードの XRES ピンと XRES/INT ピン(ピン 1 と 2) を接続させます。この設定では、CapSense コントローラーの XRES ピンが ISSP コネクタ J3 のピン 3 に接続されます。
- ヘッダーJ2 にジャンパーを嵌め、CY3280-BSM ボードのピン GND と SHILED (ピン 2 と 3) を接続させます。この設定により、CY3280-BSM ボード上のハッチパターンが接地されます。
- MiniProg1/MiniProg3 を J3 ヘッダー上の CY3280-20x66 UCC に接続します。

10.5 回路図



注: デバイスの電力消費量を測定しながら、CY3280-20x66 ユニバーサル CapSense Controller から LED D1 と D2 を除外します。SmartSense UM を備えたデバイスの消費電力は、システムの SNR により異なります。ボードレイアウトデザインが CapSense 入門ガイドのガイドラインに沿っている場合、消費電力は最小化されます。

表 10-1.サンプルコード 10 のピン割り当て

コンポーネント	ピン
ボタン 0	P0[5]
LED	P0[1]
C_{MOD}	P0[3]

10.6 ソフトウェアの準備

10.6.1 ツール

- PSoC® Designer (バージョン 5.2 以降)
- PSoC Programmer (バージョン 3.13 以降)

10.6.2 ユーザーモジュール

以下の表は、このコード表で使用されているユーザーモジュール 、および各ユーザーモジュールが占有するハードウェア リソースを一覧にします。

ユーザーモジュール	ハードウェア リソース	ユーザーモジュール名
SmartSense	CapSense ブロック、タイマー1 (デフォルト)	SmartSense
タイマー16	タイマー0	Scan_Timer

10.6.3 ユーザーモジュール パラメーター、グローバル リソース

プロジェクトの `ReadMe.txt` ファイルでは、このサンプルコードで使用されるユーザーモジュールのパラメーター設定について説明しています。また、グローバル リソースのリストが含まれています。

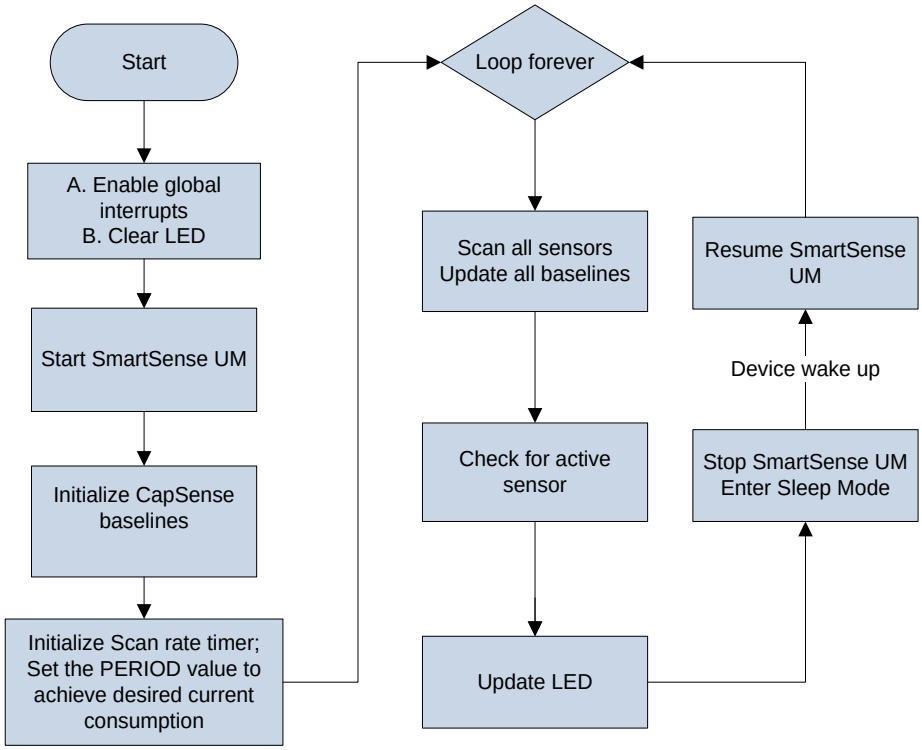
10.7 動作

リセットすると、プログラムはデバイス コンフィギュレーションからデバイスにすべてのハードウェア設定を読み出し、*main.c* を実行します。ファームウェアは以下の動作を実行します。

- SmartSense UM、ScanRate_Timer UM、LED を初期化します。
- グローバル割り込みを有効にしてから、SmartSense と ScanRate_Timer ユーザーモジュールを開始します。
- ファームウェアはボタンが押されるのを検出すると、LED をオンに切り替えます。ボタンを放すと、LED を消灯させます。この動作は、デバイスの機能を確認するために実行します。
- デバイスがスリープモードに入っても、ファームウェアは継続的にセンサーをスキャンします。デバイスは、ScanRate_Timer 割り込みでスリープから復帰します。

  スキャン速度を変更するには、マクロ SCANRATE_TIMER_VALUE を変更します。これにより、ScanRate_Timer の期間が制御されます。速度が遅くなるにつれて、応答が向上します。

図 10-2. サンプルコード 10 の機能フローチャート



10.8 コードの実行

ボードにサンプル プロジェクトをプログラムしてから、この手順を使用してサンプルコードを実行します。UCC ボードのプログラミングの方法については、CY3280-20x66 キット ガイドの第 5 章を参照してください。

1. スキャン速度を変更するには、サンプルコードでマクロ SCANRATE_TIMER_VALUE を編集します。これにより、ScanRate_Timer ユーザーモジュールのパラメーター期間が変更されます。スキャン速度は、関係式により変化します。

   スキャン速度 = 周期 / 選択したクロック周波数

   ここで、周期 = SCANRATE_TIMER_VALUE
2. 電流計を使用して、250、500、750、1000ms で異なるスキャン速度の電流を測定します。以下のグラフと表は、異なる動作電圧の異なるスキャン速度における消費電力を示します。

図 10-3. 変化するスキャン速度と電圧における平均電流

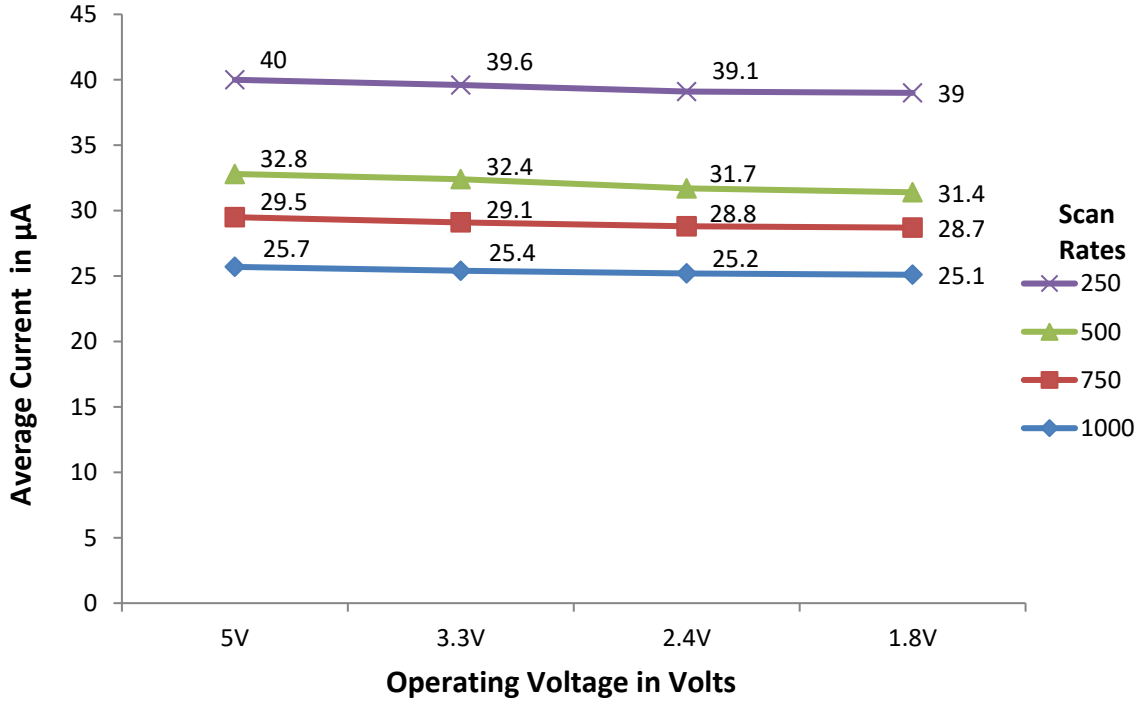


表 10-2. 変化するスキャン速度と電圧における平均電流

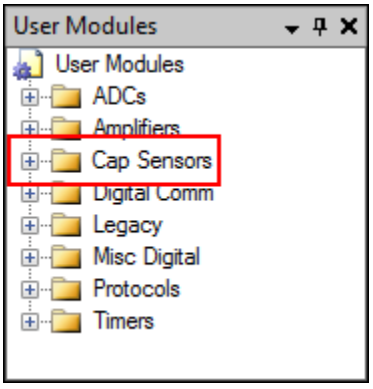
スキャン速度 (ms)	平均電流 (µA)			
	5V	3.3V	2.4V	1.8V
1000	25.7	25.4	25.2	25.1
750	29.5	29.1	28.8	28.7
500	32.8	32.4	31.7	31.4
250	40	39.6	39.1	39

付録

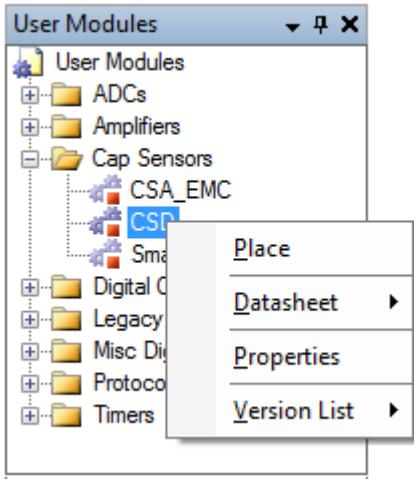
CSD ユーザーモジュールの設定

この設計ガイドでプロジェクトを修正するには、PSoC® Designer ソフトウェアで CapSense ユーザー モジュール (UM) を設定する必要があります。UM を設定すると、あるデバイスから他のデバイスにプロジェクトを移行することができます。すべての CapSense UM は、同じようなプロセスを使用して設定します。以下の例では、CSD の設定にソフトウェア ウィザードを使用します。

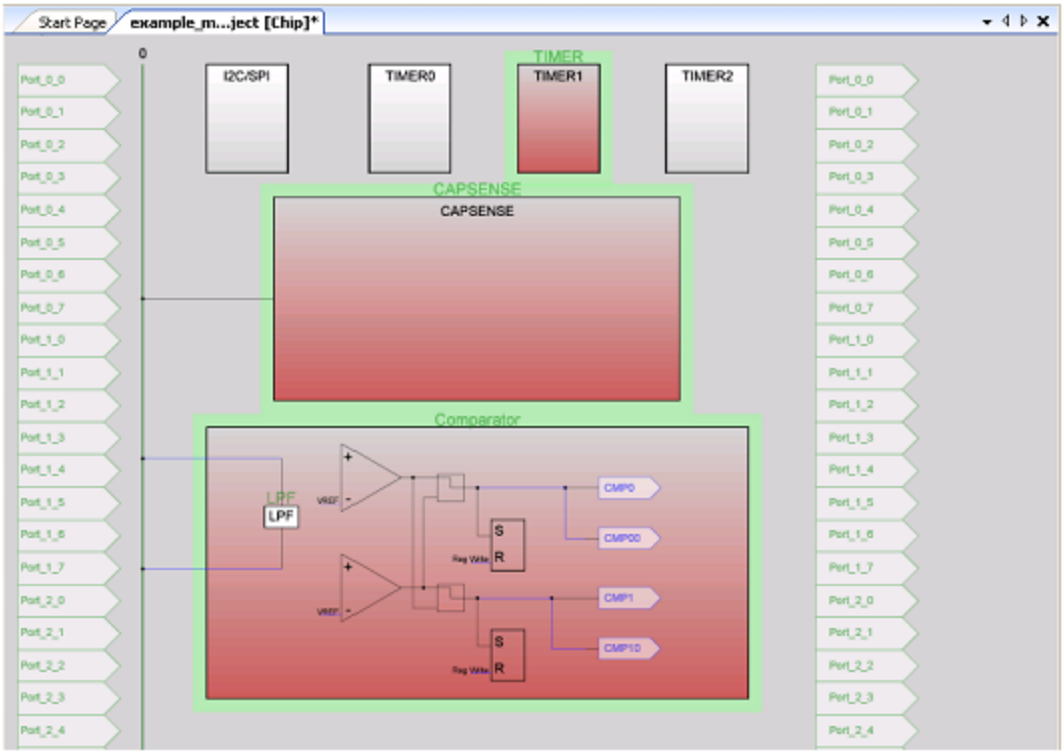
1. PSoC Designer の **User Modules** ウィンドウで、**Cap Sensors** フォルダを展開します。



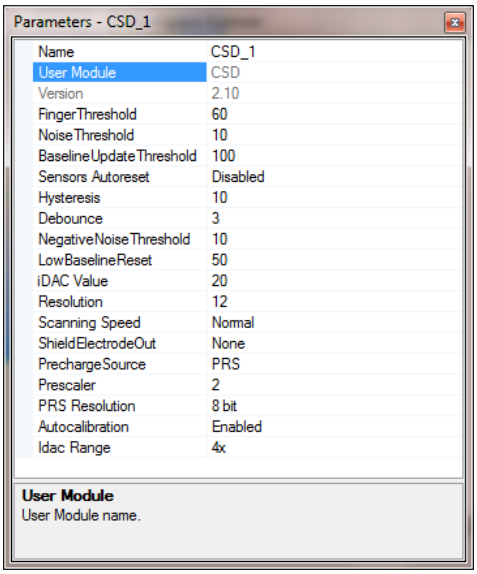
2. **Cap Sensors** フォルダで **CSD** を右クリックしてから、**Place** を選択します。



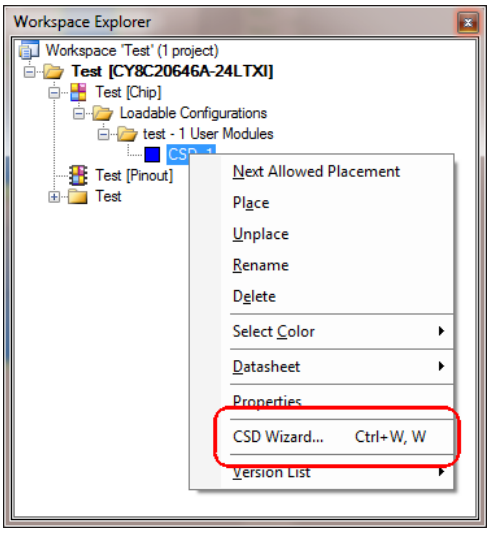
Designer ツールでは、チップ エディタの空きブロックにユーザーモジュールを配置します。



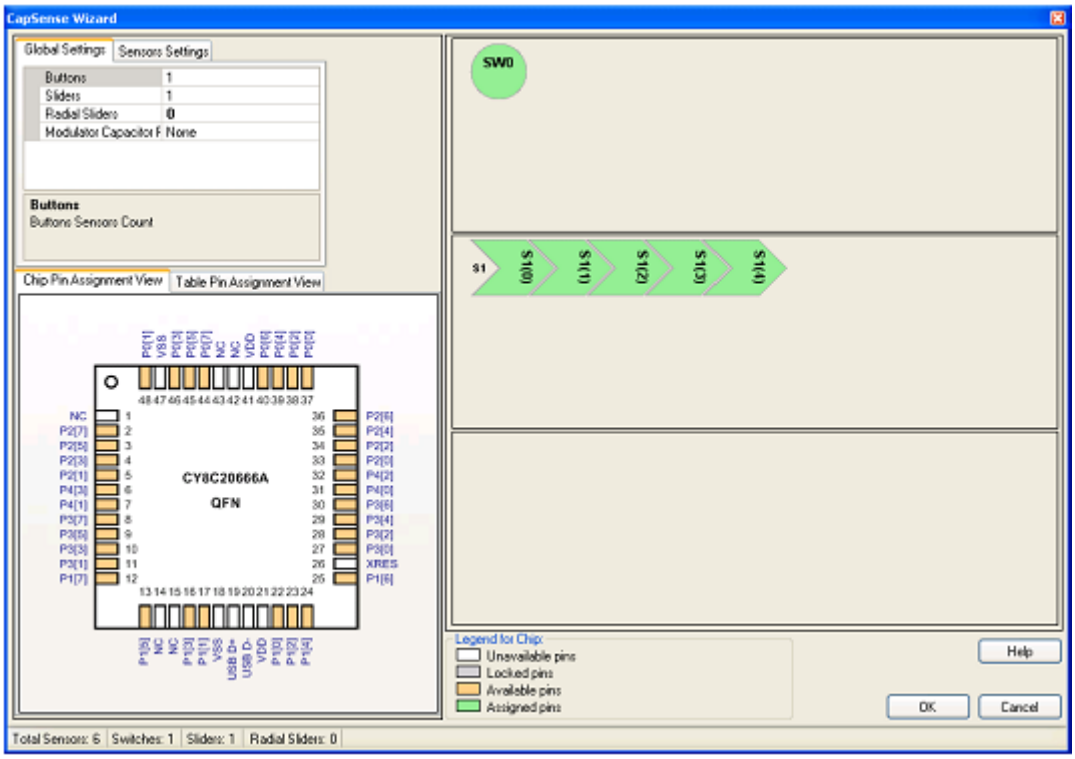
3. 以下のスクリーン ショットに示すように、CSD_1 プロパティを配置します。



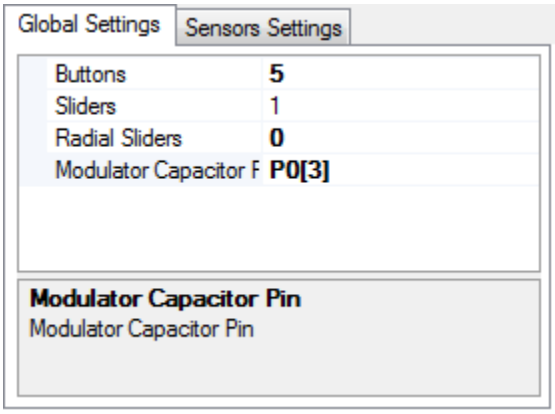
4. CSD ユーザーモジュールを右クリックしてから、**CSD Wizard** を選択します。
ウィザードを使用して、ピンをセンサーに割り当てます。



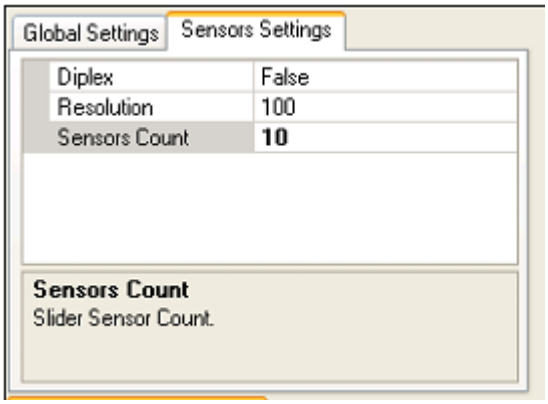
CapSense ウィザードが表示されます。



5. **CSD Wizard** ウィンドウを使用して、**Global Settings** を設定します。
   スクリーンショットの設定は、ボタンを 5 個とスライダーを 1 つ持つプロジェクトです。**Global Settings** を使用して、プロジェクトに必要なセンサー数を設定します。



6. **CSD Wizard** のスライダーをクリックすると、センサーの設定数が表示されます。以下の図のように、**Sensor Settings** を設定します。



7. 特定のピンにセンサーを割り当てるには、センサー ブロックからピン割り当てウィンドウにある必要なピンをクリックしてドラックします。SW0 をピン P1 [6]にドラッグ アンド ドロップします。センサーのピンの割り当ては、テーブル ピン割り当て表示 (図 10-5) またはチップピン割り当て表示 (図 10-4) で行えます。

図 10-4. センサーをピンに割り当てる－チップピン割り当て

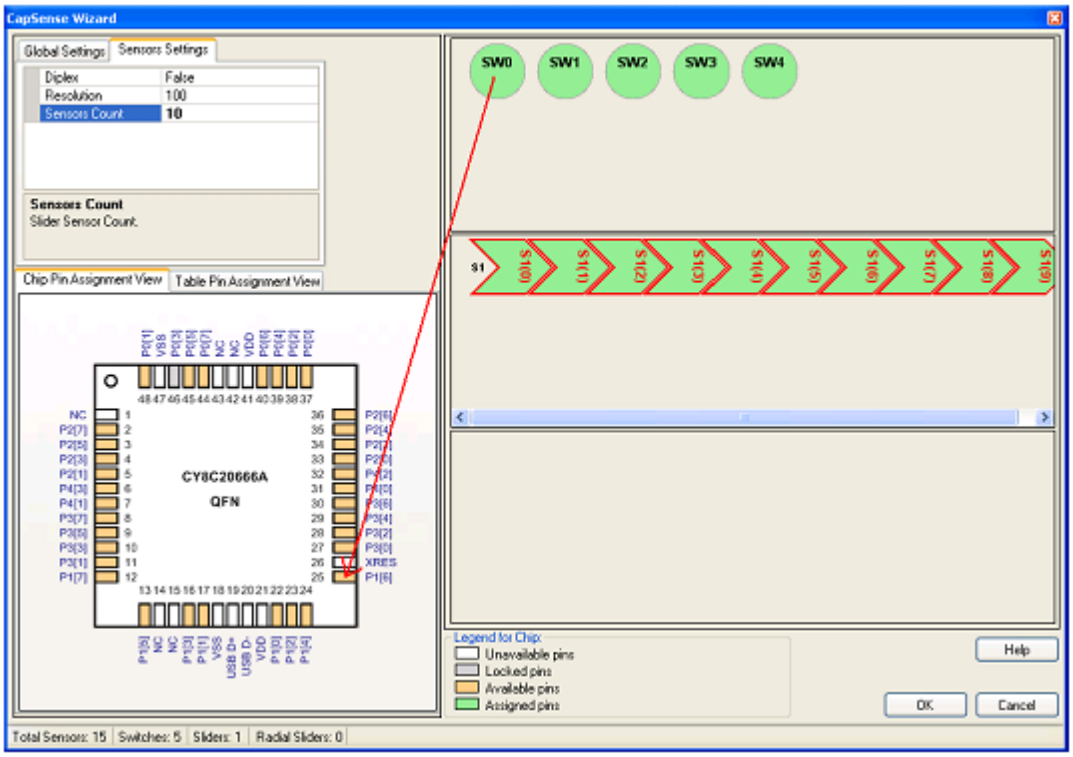
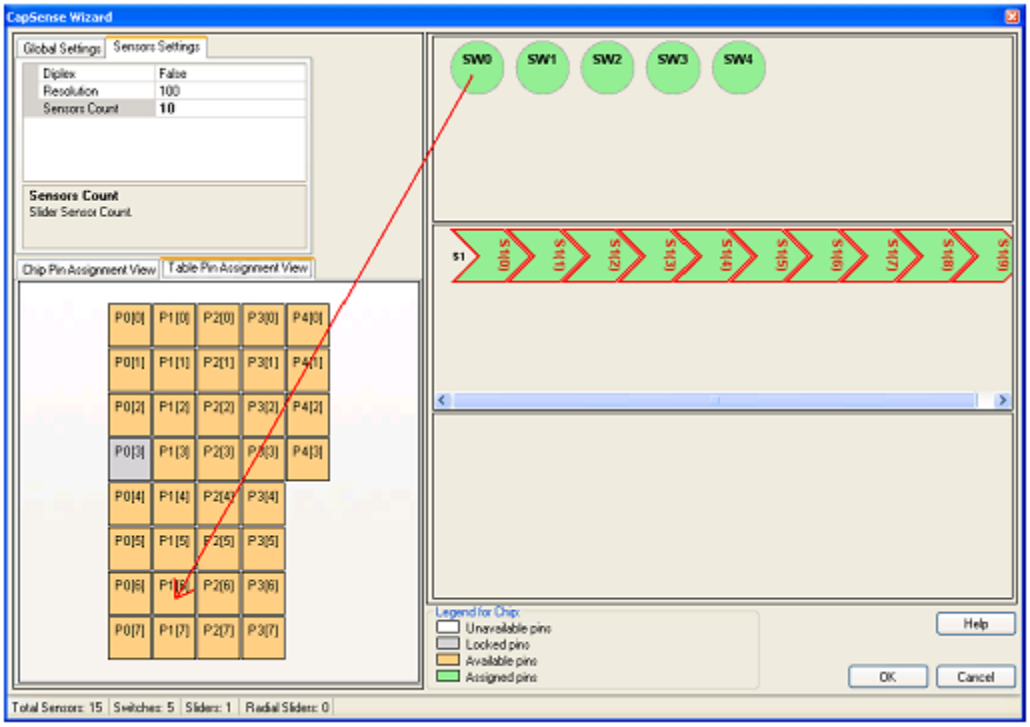


図 10-5. センサーをピンに割り当てる－テーブル ピン割り当て



8. 同様に、他のセンサーをデバイスのピンに割り当てることが可能です。**OK** をクリックして終了します。

他の CapSense デバイスへのサンプルコードの移行

エンド アプリケーションに適切なデバイスを選択するには、「Getting Started with CapSense Guide」を参照してください。本書に記載されているサンプルコードはすべて、表 1 にあるように、特定のデバイス向け (対象キット) です。サンプルコードを他の CapSense デバイスで使用するには、プロジェクトをそのデバイスに移行 (または移植) する必要があります。

移行の前に、それぞれの CapSense デバイスでサポートされている CapSense ユーザーモジュールを理解します。「Getting Started with CapSense Guide」の表 5-1 で、CapSense 製品ファミリをサポートする CapSense ユーザーモジュールの対応表を示しています。

以下のセクションでは、あるデバイスから別のデバイスにプロジェクトを移植する手順を順を追って説明します。

SmartSense

SmartSense UM は、CY8C20xx6A および CY8C21x34B デバイスファミリでサポートされています。詳細については、サンプルコード 1: SmartSense 付きボタン スライダー (CY8C20xx6A CapSense コントローラー)を参照してください。現在、当社では CY8C21x34B 部品のキットを提供していませんが、ユーザー環境で SmartSense と CY8C21x34B デバイスを一緒に使用することが可能です。CY8C21x34B のキットを希望する場合は、電子メールで capsense@cypress.com へお問い合わせください。

CapSense コントローラーとの通信

サンプルコード 2: I2C によるホスト通信(CY8C20xx6A CapSense コントローラー使用) およびサンプルコード 3: UART によるデータ送信 (CY8C20xx6A CapSense コントローラー使用)では、CapSense コントローラーCY8C20xx6A とホスト プロセッサを結びつけるための I²C と UART プロトコルの使用を説明します。これらのプロジェクトを他の CapSense コントローラーに移動するのは簡単で、それほど修正を必要としません。

サンプルコード 3 を他のデバイスに移植

1. 任意の対象デバイスで新プロジェクトを作成します。
   新プロジェクトを作成する手順は、それぞれの UCC キットガイドに記載されています。
2. UART UM を CapSense UM (CSD、CSA、CSA_EMC または SmartSense) と配置します。
3. 動作の信頼性を高めるために、CapSense UM を必要に応じてサンプルコード 7、サンプルコード 8、またはサンプルコード 9 に記載されている方法で調整します。この手順では、SmartSense は必要ありません。
4. 必要に応じて、CapSense と UART ユーザーモジュールを設定します。
   UART UM については、サンプルコード 3 で使用したのと同じ設定を使用できます。
5. Workspace Explorer で、既存の *main.c* ファイルを開きます。既存の *main.c* の内容をサンプルコード 3 プロジェクトの *main.c* ファイルの内容と置き換えます。
6. サンプルコード 3 フォルダから、*display.c* ファイルと *display.h* ファイルをプロジェクト フォルダの保存場所に追加します。
7. 3.7 コード サンプルの実行に従って、プロジェクトを実行します。
   MultiChart ツールでプロットしたデータを表示するには、「CapSense データを MultiChart でプロットする」を参照してください。

CY8C24x94 デバイスで I²C を使用

I²C プロトコルは、本書に記載されているサンプルコードのチューニングでデータ転送に使用されます。特定のデバイスで I²C を使用するには、サンプルコード 7、サンプルコード 8、およびサンプルコード 9 プロジェクトを参照してください。CY8C24x94 デバイスで I²C 通信を使用するには、以下の手順に従ってください。

1. CY8C24x94 を対象デバイスとして新しいプロジェクトを作成します。
   CY3280-24x94 UCC キットで新しいプロジェクトを作成するには、このキットのガイドを参照してください。
2. I²C ユーザーモジュールと CSD ユーザーモジュールをプロジェクトに配置します。
3. 動作の信頼性を高めるために、サンプルコード 7 に記載されている方法で CSD ユーザーモジュールを調整します。
4. 必要に応じて、CapSense と I²C ユーザーモジュールを設定します。
   I²C UM については、サンプルコード 2 で使用したのと同じ設定を使用できます。

5. サンプルコード 2 プロジェクトから *main.c* ファイル、*display.c* ファイル、および *display.h* ファイルをプロジェクトに追加します。
6. Workspace Explorer で、既存の *main.c* ファイルを開きます。既存の *main.c* の内容をサンプルコード 2 プロジェクトの *main.c* ファイルのコンテンツと置き換えます。
7. サンプルコード 2 フォルダから、*display.c* ファイルと *display.h* ファイルをプロジェクト フォルダの保存場所に追加します。
8. 2.7 サンプルコードの実行に従って、プロジェクトを実行します。
9. I²C を経由して CapSense データを表示するには、サンプルコード 1 プロジェクト フォルダの BCP コンフィギュレーション ファイル フォルダをプロジェクトの保存場所にコピーしてから、1.8 I2C を経由した CapSense データの読み出しを完了します。

絶対センサー静電容量測定

「サンプルコード 4 ― 絶対センサー静電容量の測定 (CY8C20xx6A CapSense コントローラー使用)」、「サンプルコード 5 ― 絶対センサー静電容量の測定 (CY8C21x34/B CapSense コントローラー使用)」、「サンプルコード 6 ― 絶対センサー静電容量の測定 (CY8C20x34 CapSense コントローラー使用)」は、CY8C20xx6A、CY8C21x34/B および CY8C20x34 デバイスの絶対センサー静電容量を測定します。CSD UM のアーキテクチャの類似性のため、CY8C24x94 デバイスには CY8C21x34/B のセンサー静電容量を計算する方法が使用されます。CY8C24x94 の CSD UM に関する詳細は、CY8C24x94 の CSD データシートを参照してください。

サンプルコード 5 を CY8C24x94 Device に移植

1. CY8C24x94 を対象デバイスとして新しいプロジェクトを作成します。
   CY3280-24x94 UCC キットで新しいプロジェクトを作成するには、このキットのガイドを参照してください。
2. CSD と UART UM をプロジェクトに配置します。必要に応じて、UM を設定します。
   このプロジェクトについては、サンプルコード 5 で使用したのと同じ設定を使用できます。
3. Workspace Explorer で、既存の *main.c* ファイルを開きます。既存の *main.c* の内容をサンプルコード 5 プロジェクトの *main.c* ファイルのコンテンツと置き換えます。
4. 感度の計算方法については、*CY8C21x34/B_Calculsensitivity.xls* ファイルを参照してください。*main.c* ファイルの感度を計算値と置き換えます。
5. ハイパーターミナルに絶対静電容量を表示するには、6.7 サンプルコードの実行を行います。
   サンプルコード 7～9 は CapSense UM の調整方法を説明していますが、これはプロジェクトで使用されている UM とデバイスに固有であり、 他のデバイスでは使用できません。

関連文書

設計ガイド

- CapSense 入門
- CY8C20xx6A CapSense 設計ガイド
- CY8C20x34 CapSense 設計ガイド
- CY8C21x34/B CapSense 設計ガイド

データシート

- CSD データシート
- CSA_EMC データシート
- SmartSense データシート

アプリケーションノート

- AN2397 – CapSense Data Viewing Tools

キット ガイド

- CY3280-20x66 ユニバーサル CapSense コントローラー キット ガイド
- CY3280-20x34 ユニバーサル CapSense コントローラー キット ガイド
- CY3280-21x34 ユニバーサル CapSense コントローラー キット ガイド

略語

略語	定義
AN	Application Note (アプリケーションノート)
API	Application Programming Interface (アプリケーション プログラミングインターフェース)
BSM	Simple Button Module (シンプル ボタン モジュール)
CE	Code Example (サンプルコード)
COM	Communication (通信)
CSA	CapSense Successive Approximation (CapSense 逐次比較方式)
CSD	CapSense Sigma Delta (CapSense シグマ デルタ方式)
EMC	Electromagnetic Compatibility (電磁環境適合性)
PC	Personal Computer (パーソナル コンピューター)
SLM	Linear Slider Module (リニアスライダー モジュール)
UART	Universal Asynchronous Receiver/Transmitter (ユニバーサル非同期レシーバ／トランスミッター)
UM	User Module (ユーザーモジュール)
USB	Universal Serial Bus (ユニバーサル シリアル バス)

改訂履歴

版	ECN 番号	発行日	変更者	変更内容
**	3895943	01/31/2013	HZEN	これは英語版 001-74590 Rev. *A を翻訳した日本語版 001-85884 Rev. **です。
*A	4722941	05/04/2015	HZEN	これは英語版 001-74590 Rev. *D を翻訳した日本語版 001-85884 Rev. *A です。
*B	5077313	01/08/2016	HZEN	これは英語版 001-74590 Rev. *E を翻訳した日本語版 001-85884 Rev. *B です。
*C	6498147	03/01/2019	SSAS	これは英語版 001-74590 Rev. *F を翻訳した日本語版 001-85884 Rev. *C です。