AN2397

PSoC® 1 および CapSense®コントローラー – CapSense データ監視ツール

著者: Kurian Polachan

関連製品ファミリ: CY8CMBR3xxx、CY8CMBR2xxx、CY8C201xx、CY8C20x34、CY8C20xx6A/AS/H、CY8C20xx7/S、CY8C21x34/B、CY8C24x94、CY8C22x45

ソフトウェア バージョン: Bridge Control Panel 1.12

関連アプリケーション ノート: AN49943、AN42137

さらにサンプル コードをお求めでしょうか？以下のとおり対応いたします。

PSoC のサンプル コード リストにアクセスするには、PSoC 1 および CapSense サンプル コード ウェブページをご覧ください。PSoC ビデオ ライブラリはここからご覧ください。

AN2397 は、I²C または UART インターフェースを使って PSoC 1 と CapSense コントローラーから CapSense データを監視する方法を説明します。本アプリケーション ノートで記述されているツールを使用し、CapSense チューニングやデバッグのためにリアルタイムのセンサー データを表示およびログできます。

目次

1 はじめに 2
2 CapSense リソース 2
2.1 PSoC Designer 2
2.2 サンプル コード 3
2.3 テクニカル サポート 5
3 正しい方式の選択 5
4 セットアップ 6
4.1 I²C による監視 6
4.2 UART による監視 6
5 CapSense データ監視の方法 8
5.1 プログラマブル コントローラー向けの BCP 付き I²C 8
5.2 プログラマブル コントローラー向けの BCP 付き UART 11
5.3 CY8CMBR2110 と CY8CMBR3xxx コントローラー向けの EZ-Click 付き I²C 15
5.4 CY8CMBR20xx コントローラー向けの BCP 付き UART 19
6 CY3240-I2USB ブリッジによる UART-USB ブリッジの有効化 22
6.1 CY3240-I2USB から UART-USB ブリッジの構成 22
6.2 UART-USB ドライバーのインストール 22
7 用語集 25
8 まとめ 30
改訂履歴 31

1 はじめに

CapSense の設計中に、チューニングとデバッグの際に raw カウントやベースライン、差分カウントなどの CapSense センサー データを監視する必要があります。

本文書を参考して CapSense センサー データの表示およびロギングに最適なツールを選択できます。サポートされる 2 つのインターフェースは I²C および UART です。本文書を読む前は、CapSense のセンシング技術を身につける必要があります。CapSense の一般的な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。

設計に使用するデバイスに応じて以下のリソースを選択してください。

- **PSoC 1 と CapSense コントローラー**: 本アプリケーション ノートの「正しい方式の選択」を参照してください。
- **PSoC 3 または PSoC 5LP:** PSoC 3 と PSoC 5LP CapSense デザイン ガイド内の「Tuner GUI」セクションを参照してください。
- **PSoC 4:** PSoC 4 CapSense デザイン ガイド内の「Manual Tuning Process」セクションを参照してください。

2 CapSense リソース

サイプレスは、サイプレス CapSense コントローラー ウェブサイトに大量のデータを掲載しており、ユーザーがデザインに対して適切な Capsense デバイスを選択し、迅速に効率的にデバイスをデザインに統合するための手助けをしています。CapSense と利用可能なデザイン リソースの詳細については、アプリケーション ノート AN64846「*Getting Started with CapSense*」を参照してください。

以下は Capsense のリソースの要約です。

- **概要:** CapSense ポートフォリオ – AN64846 の「*CapSense Product Portfolio*」セクションおよび CapSense ロードマップを参照してください。
- **製品セレクタ:** AN64846 の「*CapSense Selector Guide*」セクションを参照してください。PSoC Designer にはデバイス選択ツールも含まれています。
- **データシート**: CapSense デバイス ファミリ向けの電気的仕様を説明します。
- **アプリケーション ノート**および**サンプル コード**: 基本レベルから上級レベルまでの幅広いトピックを提供します。多くのアプリケーション ノートはサンプル コードを含みます。
- **テクニカル リファレンス マニュアル (TRM)**: Capsense デバイスの内部アーキテクチャの詳細を説明します。
- **開発キット:**
  - CY3280-MBR3 CapSense MBR3 評価キット: このキットは CapSense MBR3 デバイスの機能を紹介する目的で設計されています。MBR3 デバイスはレジスタ設定可能なデバイスであり、ボタンや近接センシング、耐水性など多くの機能に対応しています。
  - CY3280-BK1 ユニバーサル CapSense コントローラー – 基本キット 1:プログラマブル CapSense コントローラーの評価ができます。このキットは CY8C20x34 と CY8C21x34 デバイス用の評価基板を含みます。
  - CY8CKIT-024 CapSense 近接シールド: CapSense 近接アプリケーションの評価および開発ができます。シールドは、サイプレスのあらゆる Pioneer 開発プラットフォームで動作できるよう設計されています。
- MiniProg1 および MiniProg3 デバイス: フラッシュ メモリのプログラミング用のインターフェースを提供します。

2.1 PSoC Designer

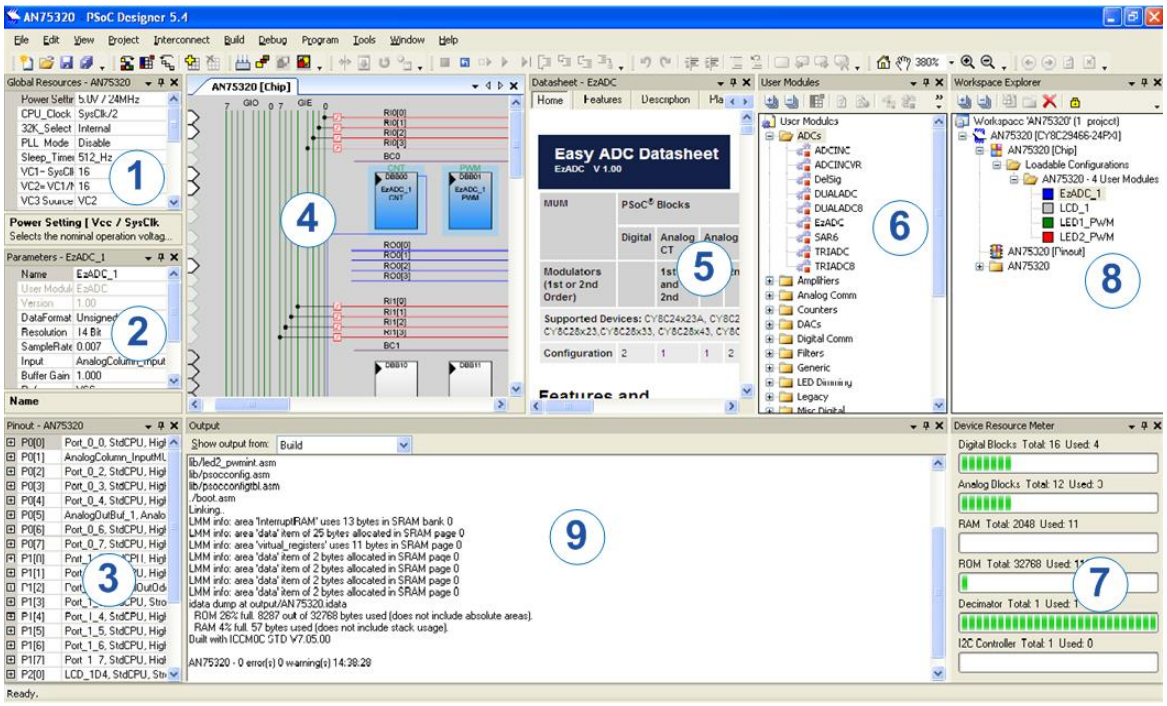
PSoC Designer は無料の Windows ベースの統合設計環境 (IDE) です。ドラッグ&ドロップの設計環境で特性化済みのアナログおよびデジタル ペリフェラルのライブラリを使用してアプリケーションを開発します。また、API ライブラリ上の動的生成が行えるコードを活用して、設計をカスタマイズすることもできます。図 1 は PSoC Designer ウィンドウを示します。
注: これは、デフォルト画面ではありません。

1. **Global Resources (グローバル リソース):** すべてのデバイス ハードウェアの設定
2. **Parameters (パラメーター):** 選択したユーザー モジュール**のパラメーター**
3. **Pinout (ピン配置):** デバイスのピンに関する情報

4. **Chip-Level Editor (チップ レベル エディタ):** 選択したチップで使用可能なリソースの図
5. **Datasheet (データシート):** 選択したユーザー モジュールのデータシート
6. **User Modules (ユーザー モジュール):** 選択したデバイスのすべての使用可能なユーザー モジュール
7. **Device Resource Meter (デバイス リソース メーター):** 現行のプロジェクト コンフィギュレーションのデバイス リソース使用率
8. **Workspace (ワークスペース):** プロジェクトに関するファイルを表示するツリー レベル図
9. **Output (出力):** プロジェクト ビルドおよびデバッグ処理からの出力

注: PSoC Designer の詳細情報については、**PSoC®Designer**> **Help** > **Documentation** > **Designer Specific Documents** > **IDE User Guide** を順に選択してください。

図 1. PSoC Designer のレイアウト



2.2 サンプル コード

このウェブページには PSoC Designer ベースのサンプル コードがリストアップされています。これらのサンプル コードは、空のページの代わりに完了したデザインで始めることによって設計時間を短縮させることができます。また、PSoC Designer ユーザー モジュールがさまざまな用途にどのように利用できるかも示します。さらに、CapSense コントローラー サンプル コードのデザイン ガイドは Capsense デザインで使用される重要なサンプル コードも提供しています。

PSoC Designer に統合されているサンプル コードへアクセスするには、図 2 に示すように **Start Page** > **Design Catalog** > **Launch Example Browser** を順に選択してください。

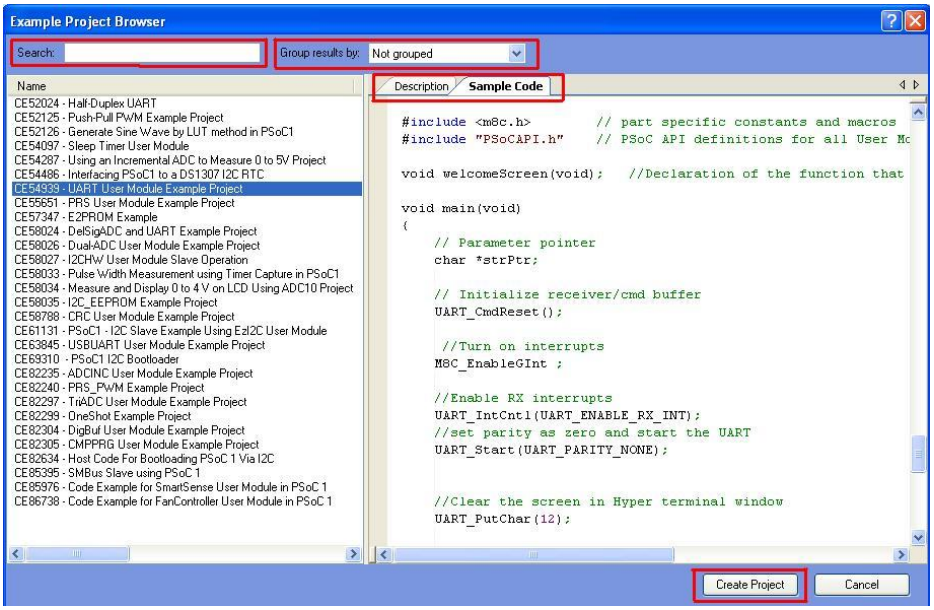
図 2. PSoC Designer 内のサンプル コード



図 3 に示す Example Projects Browser には、以下のオプションがあります。

- プロジェクトを絞り込むためのキーワード検索
- カテゴリ ベースのプロジェクト リスト
- 選択したプロジェクトのデータシートのレビュー機能 (Description タブ内)
- 選択したプロジェクトのサンプル コードのレビュー機能。このウィンドウからコードをコピーし、プロジェクトに貼り付けてコード開発時間を短縮できます。または、
- 選択したものをベースに新規プロジェクト (また必要な場合は新規ワークスペース) を作成できます。出来上がった基本的な設計を元に始めることで設計時間を短縮します。その設計をユーザーのアプリケーションに適用できます。

図 3. サンプル プロジェクトおよびサンプル コード



2.3 テクニカル サポート

ご質問には弊社のテクニカル サポート チームが対応させて頂きますので、お気軽にご連絡ください。サイプレス テクニカル サポート ページにアクセスし、お問い合わせ内容をケースとして作成し、送信してください。

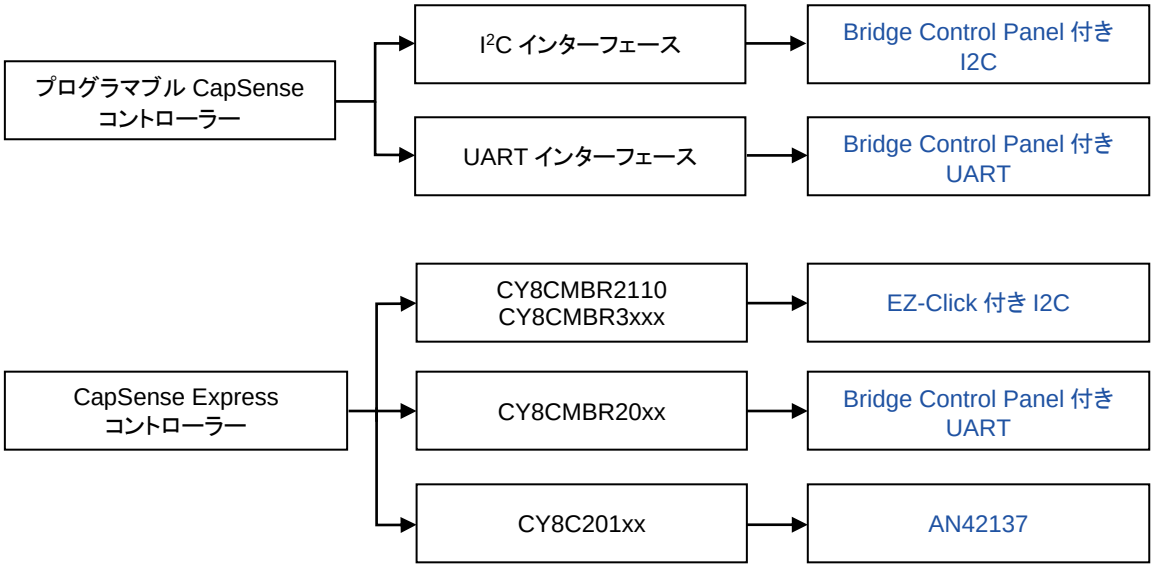
早急な対応が求められる場合には、下記のリソースをご利用ください。

- セルフ ヘルプ
- 販売事務所所在地

3 正しい方式の選択

サイプレスは、設定やプログラムが可能な多種多様な CapSense コントローラー (Capsense Express) を提供しています。通信インターフェースとデータ監視用のツールがデバイスによって異なります。図 4 には選択可能な方式を示します。リンクをクリックして関連する節へ進みます。ハードウェア セットアップは次の節で説明します。

図 4. データ監視方式の選択図



CY8C201xx ファミリでは、CapSense データ閲覧ツールは、PSoC Designer™ (5.0 SP6 版) に直接統合されています。詳細は、AN42137 を参照してください。

プログラマブル CapSense コントローラーの場合、I²C または UART は通信インターフェースとして使用されています。表 1 は、どのインターフェースを使用するか示します。

表 1. プログラマブル コントローラー向けの I2C と UART の比較

比較マトリックス	I²C	UART
必要なピン	2	1
データ損失	有	無
双方向通信サポート	有	無

注: EZ-Click は、プログラマブル コントローラーのセンサー データ読み出しをサポートしていません。

- I²C は CapSense コントローラーの 2 本の空いているポート ピン (SCL と SDA) を必要としますが、UART は 1 本だけ (TX8 ユーザー モジュールの送信ピン) が必要です。

- I²C 方式では、PC は CapSense コントローラーから非同期 (CapSense スキャンと非同期) でデータを読み出します。この場合は、PC が CapSense コントローラーからデータを読み出す速度より CapSense スキャン レートが早い場合、サンプルが失われます。UART 方式では、CapSense コントローラーからのデータは同期して PC に送信されます (データは各 CapSense スキャン後に送信されます)。この通信モードではサンプルが失われません。
- BCP は UART を介してデータを Capsense コントローラーに送信することをサポートしていません。

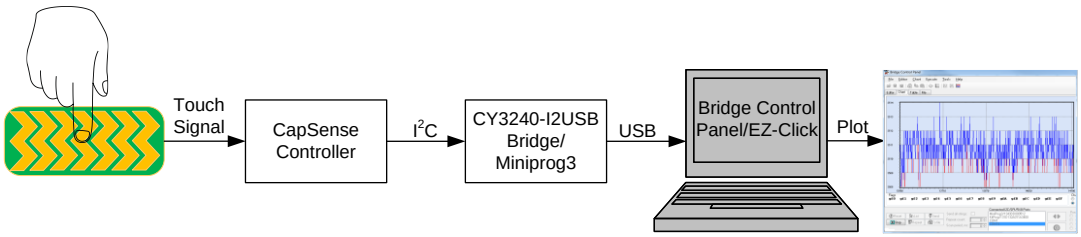
4 セットアップ

CapSense コントローラーは Capsense データを監視するために、I²C または UART インターフェースをサポートしています。ハードウェア セットアップは通信インターフェースによって異なります。CapSense デバイスがサポートするインターフェースによって適当なセットアップを選択してください。

4.1 I²C による監視

この方法では、図 5 に示すように、データは I²C インターフェースを介して CapSense コントローラーから読み出され、CY3240-I2USB[a]ブリッジ (I²C-USB コンバータ) または MiniProg3 キットを使って USB を介して PC に送信されます。PC 側では、Bridge Control Panel (BCP) または EZ-Click™ツールを使ってセンサー データを表示しログします。BCP は I²C または UART を介してデータを読み出すためにサイプレスによって提供されるツールであり、PSoC Programmer とともにインストールされます。EZ-Click は、CY8CMBR2110 と CY8CMBR3xxx デバイスのセンサー コンフィギュレーションの設定、リアル タイム センサー出力の監視や生産ライン システム診断の実行に使用されます。

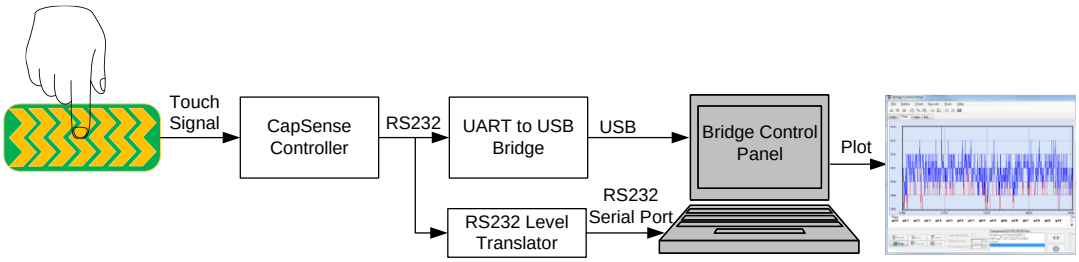
図 5.I²C インターフェースを経由した CapSense コントローラーからのデータ読み出し



4.2 UART による監視

この方法では、図 6 に示すように、データは UART／RS232 インターフェースを介して CapSense コントローラーから読み出され、USB インターフェースまたは RS232 シリアル ポートを介して PC に送信されます。PC 側では、BCP ツールを使ってを表示しログします。

図 6. UART を経由した CapSense コントローラーからのデータ読み出し



CapSense コントローラーにより送信される RS232 データは以下の方式の 1 つを使って PC に送信されます。

- UART-USB ブリッジ:「CY3240-I2USB ブリッジによる UART-USB ブリッジの有効化」を参照して CY3240-I2USB キットを使用してください。PSoc 1 デバイスに UART-USB ブリッジを実装する方法については、AN49943 を参照してください。

[a] CY3240-I2USB キットは旧式のものでもうはやサポートされていなくて、その代わりに Miniprog3 を使用します。しかし、CY3240-I2USB キットは他の CapSense キット (CY3280-BK1、CY3280-20x66、CY3218-CAPEXP1、CY3218-CAPEXP2 など) とともに提供されています。

- RS232 レベル トランスレータの使用: PC が RS232 シリアル ポートを備えている場合、データを CapSense コントローラーから PC に送信するするために RS232 レベル トランスレータを使用できます。

5 CapSense データ監視の方法

5.1 プログラマブル コントローラー向けの BCP 付き I²C

CapSense コントローラー サンプル コードのデザイン ガイドのサンプル コード 2 は、I²C スレーブ デバイスから CapSense データを読み出し、BCP ツールで表示する方法を示します。

5.1.1 ステップ A: BCP のインストール

1. BCP ツールは、PSoC Programmer とともにインストールされます。www.cypress.com/programmer からツールをダウンロードします。
2. インストール後、**Start** > **All Programs** > **Cypress** > **Bridge Control Panel [バージョン]** > **Bridge Control Panel [バージョン]** の順で進んでアプリケーションを起動します。

5.1.2 ステップ B: CapSense コントローラーでの I²C スレーブ インターフェースの実装

1. PSoC Designer では、プログラマブル CapSense コントローラーのいずれかの新規プロジェクトを作成します。
2. EzI2Cs ユーザー モジュールをプロジェクトに配置します。
3. ユーザー モジュールのパラメーターを以下のように設定します。
   - Slave_Addr = 0 (このアドレスを 0～127 のいずれかに設定可能)
   - Address_Type = Static
   - ROM_Registers = Disable (ほとんどの場合)
   - I2C Clock = 400 kHz Fast
   - I2C Pins = P1[0]-P1[1] または P1[5]-P1[7]
4. I²C 送信に必要なデータを格納する RAM バッファを定義します。定義の方法は、サンプル コード 2 を参照してください。SetRamBuffer 関数の呼び出し時にアドレスが指定されるアレイまたは構造を定義します。例:

```
struct I2C_Regs
{
    BYTE bSnsIndex; // read/write value
    BYTE bSnsMask; // read only value
    WORD wRawCount; // read only value
    WORD wBaseline; // read only value
    WORD wDiffCount; // read only value
    WORD wCentroid; // read only value
} MyI2C_Regs;
```

5. 以下の 2 行をプログラムの初期化の部分に挿入します。

```
EzI2Cs_SetRamBuffer(sizeof(MyI2C_Regs),1,(BYTE*)&MyI2C_Regs);

EzI2Cs_Start();
```

   最初の行を変更して EzI2Cs_SetRamBuffer 関数の引数を必要に応じて設定します。最初の引数は、読み出せるデータの長さを設定します。第 2 の引数は、書き込めるデータの長さを設定します。第 3 の引数は、I²C バッファのアドレスを設定します。これらの関数の詳細は、EzI2C スレーブ ユーザー モジュール データシートを参照してください。

6. 各スキャンの後、RAM バッファを更新します。例:

```
MyI2C_Regs.bSnsMask = CSD_baSnsOnMask[0];
MyI2C_Regs.wRawCount = CSD_waSnsResult[MyI2C_Regs.bSnsIndex];
MyI2C_Regs.wBaseline = CSD_waSnsBaseline[MyI2C_Regs.bSnsIndex];
MyI2C_Regs.wDiffCount = CSD_waSnsDiff[MyI2C_Regs.bSnsIndex];
MyI2C_Regs.wCentroid = CSD_wGetCentroidPos(1);
```

   EzI2Cs ユーザー モジュールは、I²C 割り込みハンドラとともにバックグランドで動作します。

   I²C スレーブが受信した最初のバイトは、データが読み書きされるバッファへのオフセットです。デフォルト オフセットは 0 です。EzI2Cs ユーザー モジュールの使用については、ユーザー モジュール データシートを参照してください。

注: 長さが 2 バイト以上の変数のデータ整合性を保証するためには、これらの変数をコードで変更する前に EzI2Cs_bBusy_Flag 変数を確認します。例:

```
EzI2Cs_DisableInt();
if(!(EzI2Cs_bBusy_Flag == EzI2Cs_I2C_BUSY_RAM_READ))
{
    MyI2C_Regs.wData++; //safely increment a 2 byte variable
}
        EzI2Cs_ResumeInt();
```

5.1.3 ステップ C: ハードウェアのセットアップ

図 5 に、この方式でどのようにコンポーネントを相互接続するかを示します。

CY3240-I2USB ブリッジを使用している場合、CapSense コントローラーの I²C ピンを CY3240-I2USB ブリッジに接続する方法については、CY3240-I2USB ブリッジ ガイドを参照してください。

MiniProg3 を使用している場合、CapSense コントローラーの I²C ピンを MiniProg3 に接続する方法については、Miniprog3 ユーザー ガイドを参照してください。

5.1.4 ステップ D: BCP による CapSense センサー データの読み出し

図 7、図 8 および図 9 に、さまざまな BCP ウィンドウのスクリーンショットを示します。アプリケーション ユーザー インターフェース、サポートされているコマンド、変数の定義、データ取得の複数のモード (Repeat モード、To File モード) などの詳細情報を示すヘルプ トピックは BCP のメニュー バーにあります。

図 7. BCP: コマンド エディタ ビュー

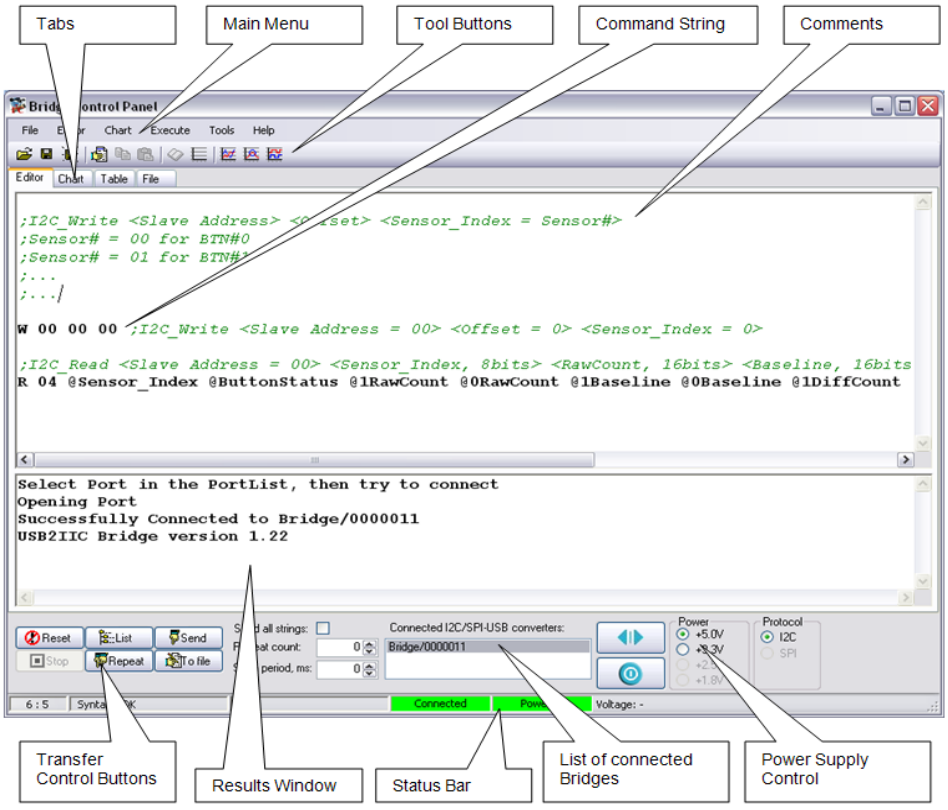


図 8.BCP: チャート ビュー (raw カウントとベースラインを表示)

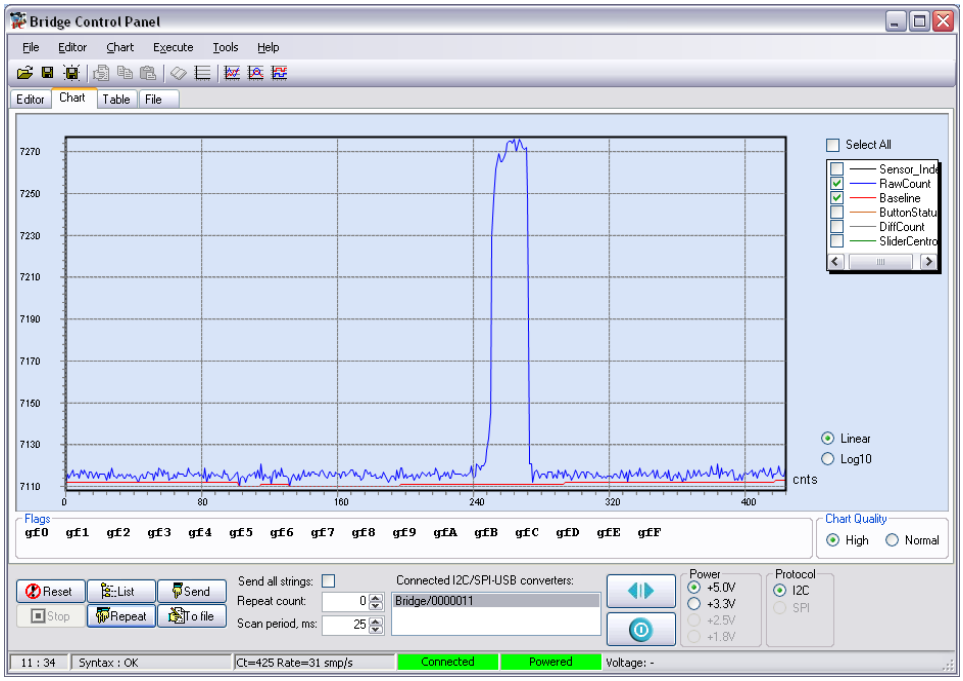
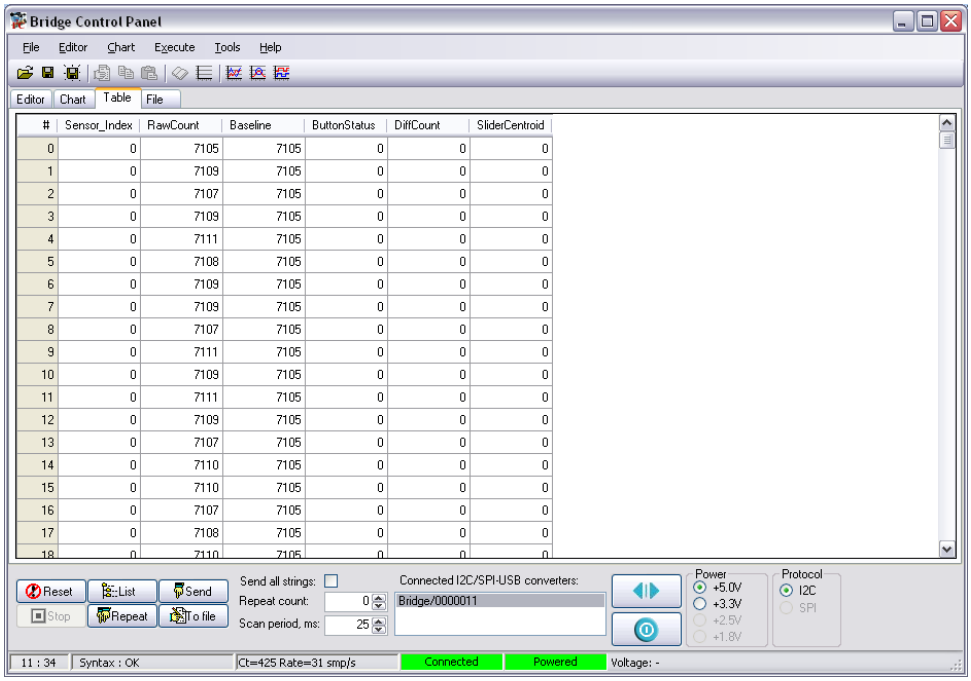


図 9.BCP: センサー データ表



5.2 プログラマブル コントローラー向けの BCP[a]付き UART

5.2.1 システム要件

この方式を適用する際は以下のシステム コンフィギュレーションが必要です。

- 空いているシリアル通信 (COM／RS232) ポート[b]
- Microsoft Windows XP、7 またはそれ以降のバージョン

CY3240-I2USB キットの使用法は、「CY3240-I2USB ブリッジによる UART-USB ブリッジの有効化」を参照してください。PSoc 1 デバイスに UART-USB ブリッジを実装する方法は、AN49943 を参照してください。

この方式では、以下の手順に従って CapSense データを閲覧します。

5.2.2 ステップ A: BCP のインストール

BCP ツールは、PSoC Programmer とともにインストールされます。www.cypress.com/programmer からツールをダウンロードします。

5.2.3 ステップ B: CapSense コントローラーからのデータ送信

プログラマブル デバイスの場合、PSoC Designer プロジェクトで以下のステップに従います。

1. TX8SW ユーザー モジュールを PSoC Designer プロジェクトに配置します。
2. ユーザー モジュールのパラメーターを以下のように設定します。
   - **Port Pin** = RS232 シリアル送信ピンを選択します。
   - **Baud Rate** = ボーレートの一覧から 1 つを選択します。オプションには、115200、57600、38400、19200、9600、4800、2400、1200bps があります。
   - **Parity** = NONE
   - **Stop Bits** = ONE
   - **Data Bits** = 8
3. 以下の文をプログラムの初期化の部分に挿入します。

```
TX8SW_Start();
```

4. ここで、BCP がデータを解析できるように特定の CapSense データ パケット構造を送信する方法を説明します。CapSense スキャンが終わった後、所与のコードをメイン ループに挿入する必要があります。BCP アプリケーションは、パケットに分けられたデータを受け取ります。

パケット 0	パケット 1	…	パケット N	…

表 22 に示すように、各パケットはヘッダ、データ要素、テールから構成されます。「ステップ D: BCP による CapSense センサー データの読み出し」に記述したように、パケットを正確に解析するために、BCP アプリケーションに入力された RX8 コマンドは PSoC によって送信されたパケット構造と一致する必要があります。表 2 に示されるパケット形式を前提とし、それに従ってコードを書きます。

[a] UART を介してデータを監視するために使用される MultiChart はもはやサポートされなくなり、その代わりに Bridge Control Panel が使用されます。

[b] UART-USB ブリッジがシリアル (RS232) データを読み出すために使用される場合、この要件は必須ではありません。

表 2. データ パケット構造

パケット要素	内容
ヘッダ	0x0D 0x0A
データ	raw カウント 0 raw カウント 1 … ベースライン 0 ベースライン 1 … 信号 0 信号 1 …
テール	0x00 0xFF 0xFF

5. ヘッダにより、BCP は入力バイト ストリームをパケットに分けます。ヘッダは、データの一部ではない任意のバイト シーケンスを含むことがあります。ここで使用するヘッダは、キャリッジ リターン (0x0D) とライン フィード (0x0A) の 2 バイトから構成されます。ヘッダを送信するには、標準的な TX8 ユーザー モジュール関数を呼び出します。

```
TX8SW_PutCRLF(); // Send Header
```

ヘッダの次にはデータが続きます。各センサーのデータ部分は 3 つの 16 ビット ワード (raw カウント、ベースライン、差分カウント) から構成されます。各ワードは BCP の符号なし整数の変数[a]で表されます。データ ワードは、表 2 に従って順序付ける必要があります。次のコード シーケンスでデータ部分を (すべてのセンサー結果を正しい順で) PC に送ります。

```
TX8SW_Write((char *) (CSD_waSnsResult), CSD_TotalSensorCount*2);

TX8SW_Write((char *) (CSD_waSnsBaseline), CSD_TotalSensorCount*2);

TX8SW_Write((char *) (CSD_waSnsDiff), CSD_TotalSensorCount*2);
```

raw カウントなどの 16 ビット値の場合、MSB は最初に送信され、それに続いて LSB が送信されます。

6. パケットの終端はテールです。テールにより、BCP はパケットの長さを判定します。ヘッダと同様に、テールはデータの一部ではない任意のバイト シーケンスを含むことがあります。この場合、テールは 0x00、0xFF、0xFF の 3 バイトから構成されます。テールを送信するには、プロジェクトに次のコードを追加します。

```
TX8SW_PutChar(0x00); // Send Tail
TX8SW_PutChar((CHAR)0xFF);
TX8SW_PutChar((CHAR)0xFF);
```

5.2.4 ステップ C: ハードウェアのセットアップ

図 6 に、この方式でどのようにコンポーネントを相互接続するかを示します。CY3240-I2USB ブリッジを UART-USB ブリッジとして使用する場合、以下のルールを適用します。

- CapSense コントローラーのシリアル送信ピンを CY3240-I2USB ブリッジの I2C_SDA ピンに接続します。
- CapSense コントローラーのグランドを CY3240-I2USB ブリッジの GND ピンにワイヤで接続します。

CY3240-I2USB のピン配置の詳細は、CY3240-I2USB ブリッジ ユーザー ガイドを参照してください。

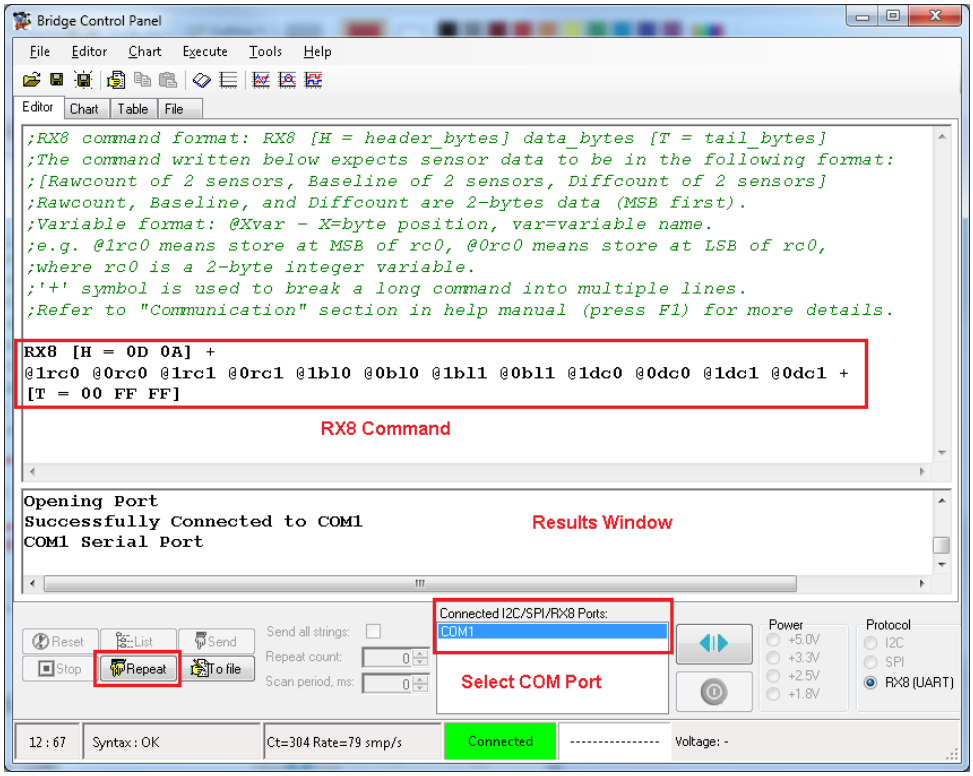
[a] 各データは BCP での名付けられた変数で表されます。例えば、rc0 はセンサー0 の raw カウントを示します。BCP はそのような変数を最大 32 種までサポートしています。変数のデータ型は符号なしまたは符号付きの byte／int／long int／float です。詳細は、BCP のヘルプ マニュアルを参照してください。

5.2.5 ステップ D: BCP による CapSense センサー データの読み出し

BCP を使用して CapSense センサー データを読み出すには、以下の手順に従ってください。

1. **Start** > **All Programs** > **Cypress** > **Bridge Control Panel [バージョン]** > **Bridge Control Panel [バージョン]** の順で選択して BCP アプリケーションを起動します。図 10 に、アプリケーションの メイン ウィンドウを示します。

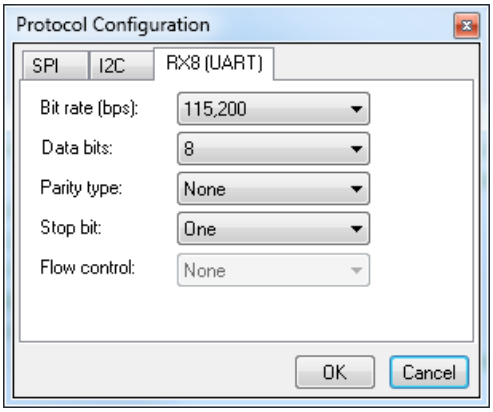
図 10. メイン ウィンドウ



2. UART プロトコルの設定を行います。

図 10 のメイン ウィンドウに示すように、 **Connected Ports** の下の必要な **COM** ポートを選択します。RX8 (UART) はプロトコルとして自動的に選択されます。次に、**Tool** メニューから **Protocol Configuration** を選択します。すると、図 11 に示す設定ダイアログ ボックスが表示されます。RX8 (UART) タブを選択して、PSoC Designer プロジェクトと一致するように設定してください。

図 11.設定ダイアログ ボックス



3. CapSense データを監視します。

   次の手順で、BCP によりデータを監視する方法を概説します。詳細は、BCP のヘルプ マニュアル (F1 を押す) を参照してください。

   a. 図 10 に示されるメイン ウィンドウから **Chart** > **Variable Settings** の順で選択し、Variable Settings のウィンドウで変数を追加します。変数を追加するために、**Active** ボックスにチェック マークを入れ、**Variable Name**(変数名) を編集し、**Type** (データ タイプ) および **Color** (色) を選択します。変数の追加が完了したら、**OK** をクリックします。**Save** をクリックして変数設定をファイルに保存し、**Load** をクリックして後でロードできます。

   図 12 に、2 つのセンサーの raw カウント (rc0、rc1)、ベースライン (bl0、bl1) および差分カウント (dc0、dc1) の変数が追加される Variable Settings ウィンドウを示します。

   図 12. Variable Settings ウィンドウ

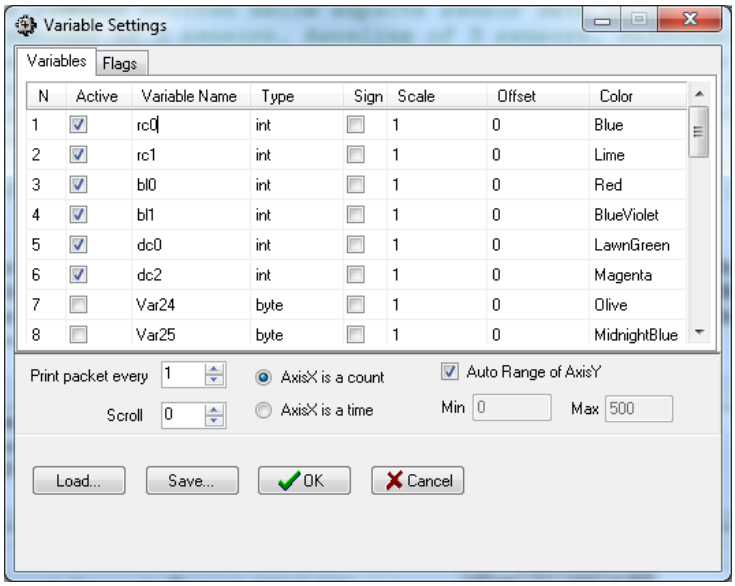


   図 10 に示すように、メイン ウィンドウのコマンド エリアで次の RX8 コマンドを入力します。

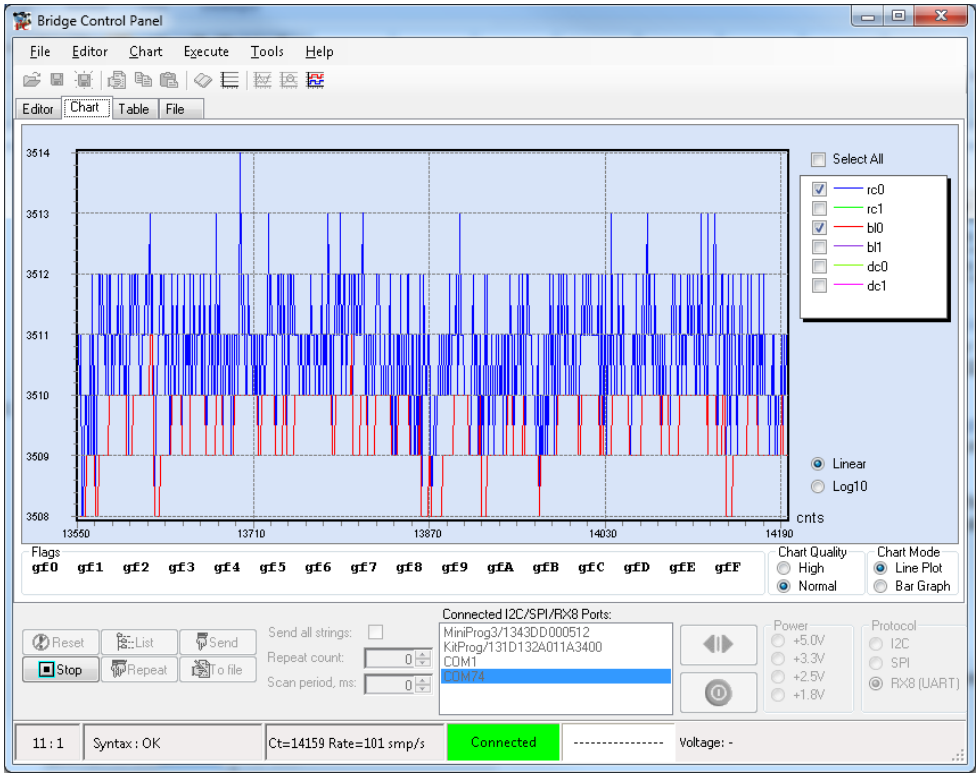
   RX8 [H = 0D 0A] +
   @1rc0 @0rc0 @1rc1 @0rc1 @1bl0 @0bl0 @1bl1 @0bl1 @1dc0 @0dc0 @1dc1 @0dc1 +
   [T = 00 FF FF]
   コマンドの構成を以下のとおりに説明します。「+」記号を使用してコマンドを複数の行に分割します。

RX8	UART データを期待するために BCP を指示するコマンド
[H = 0D 0A]	ヘッダ バイト
@1rc0….@0dc1	データ バイト
[T = 00 FF FF]	テール バイト

   BCP はデータをバイト ストリームと見なします。したがって、必要に応じてマルチバイト変数を複数バイトに分割し、また分割した各バイトを@Xvariablename として表示する必要があります (ここで、X は変数でのバイト位置を示す数字です)。例えば、センサー0 の raw カウント (2 バイト) の MSB と LSB はそれぞれ@1rc0 と@0rc0 として示されます。MSB が最初に送信されるため、コマンドでは@1rc0 は@0rc0 の前に現れます。

   b. RX8 コマンド全体を選択して、**Repeat** をクリックします。受信されたデータは結果ウィンドウに表示されます。次に、**Chart** タブに進んで、監視する必要のある変数を選択します。図 13 はセンサー0 の raw カウントとベースラインのグラフを示します。

図 13. Bridge Control Panel のチャート ウィンドウ



5.3 CY8CMBR2110 と CY8CMBR3xxx コントローラー向けの EZ-Click 付き I²C

EZ-Click は GUI ベースのツールであり、I²C インターフェースを介して CY8CMBR2110 と CY8CMBR3xxx CapSense デバイスを設定するために使用されます。

5.3.1 ステップ A: EZ-Click ツールのインストール

1. www.cypress.com/go/EZ-Click から EZ-Click ソフトウェアをダウンロードして、インストールします。
2. インストールの後、**Start** > **All Programs** > **Cypress** > **EZ-Click 2.0**> **EZ-Click 2.0** の順で選択してデフォルトの場所から EZ-Click ツールを起動します。

5.3.2 ステップ B: CY8CMBR2110／CY8CMBR3xxx コントローラーの設定

以下のタスクの詳細は、EZ-Click 2.0 ユーザー ガイドを参照してください。

1. EZ-Click ツールで新規プロジェクトを作成します。
2. 選択した CapSense コントローラーの各種のパラメーターを指定し、コンフィギュレーション ファイルを生成します。
3. 図 14 に示すように、**Configuration** > **Select Target Device** を選択して CapSense コントローラーを設定します。
   図 15 に示すように、PC に接続しているキット (CY3280-MBR2 キットか CY3280-MBR3 キット) またはデバイスに応じて、さまざまな「**ポート (Port)**」や「**デバイス (Device)**」が選択できます。デバイスの接続先であるポートを選択します。「**Power**」と「**I²C speed**」で 1 つのオプションを選択して、**OK** をクリックします。

   注: CY3280-MBR3 キットおよび CY3280-MBR2 キットは内蔵 I²C-USB ブリッジを備えており、USB A-Mini-B ケーブルを使って PC に直接接続できます。CY8CMBR2110 または CY8CMBR3xxx デバイスはカスタム基板に実装される場合、デバイスを PC に接続するために CY3240-I2USB ブリッジまたは MiniProg3 キットを使用する必要があります。

図 14. EZ-Click でのコンフィギュレーション生成

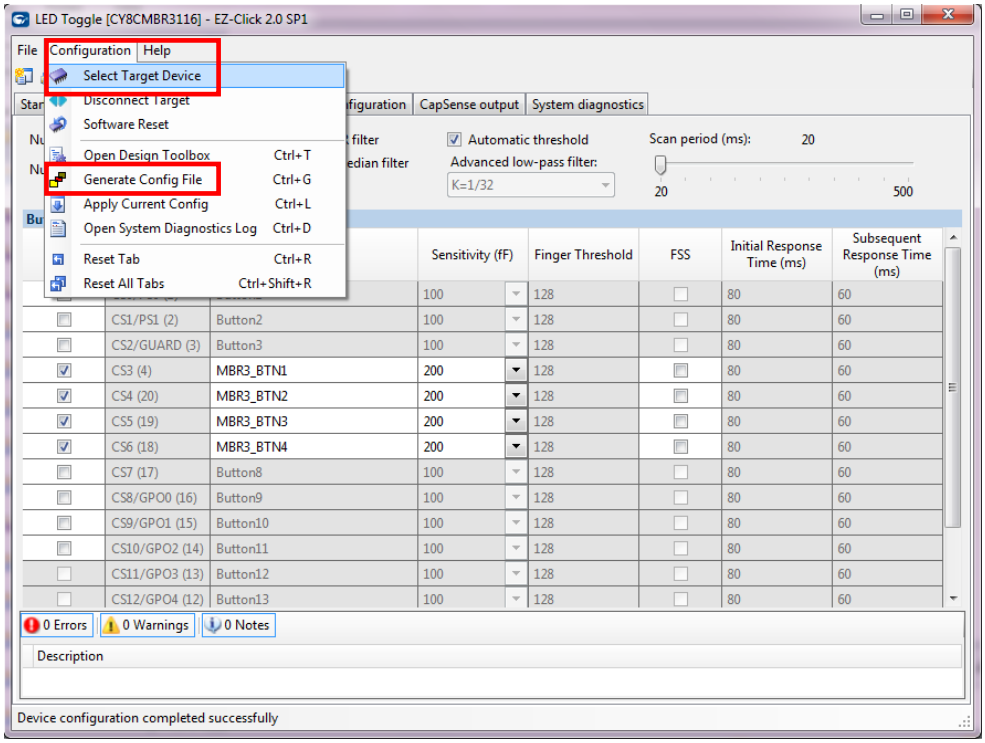
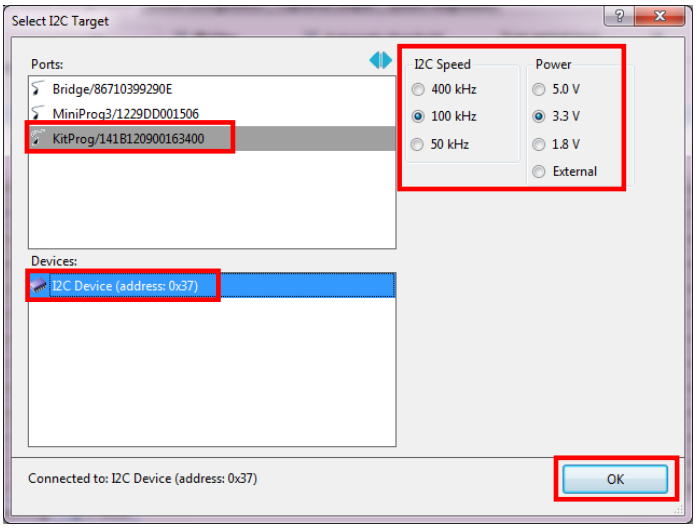
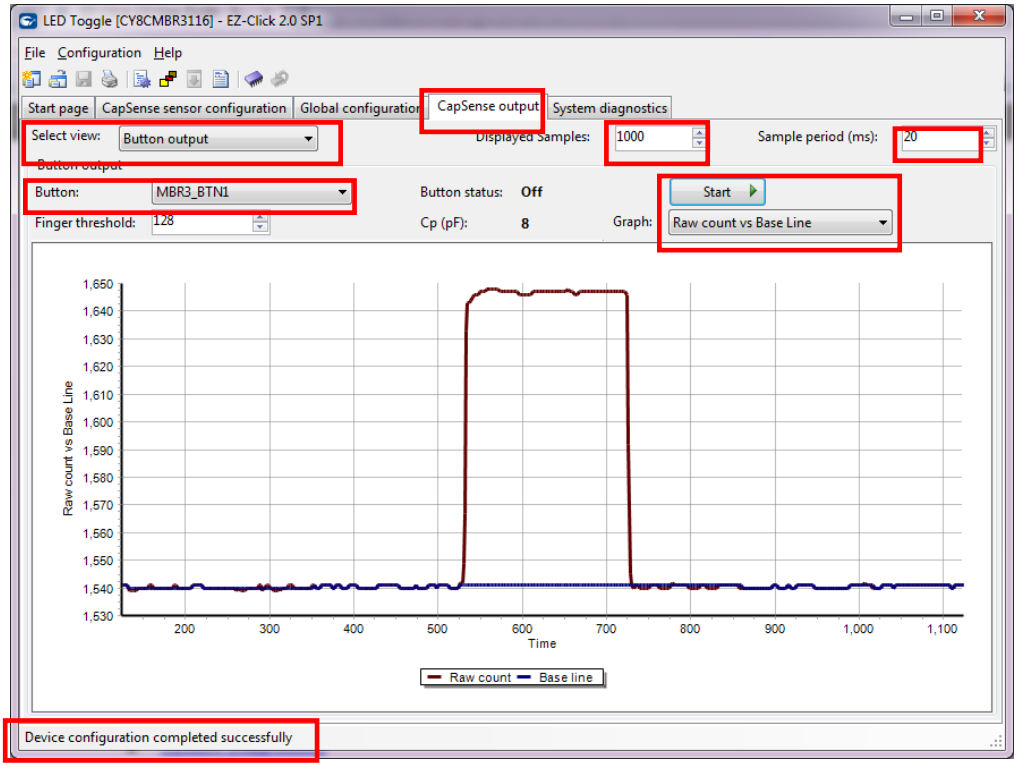


図 15. EZ-Click でのターゲット デバイスの選択



4. **Configuration**> **Generate Config File** を選択して、コンフィギュレーション ファイルを生成します。
5. **Configuration** > **Apply Current Config** を選択して、CapSense コントローラーを設定します。
6. CapSense コントローラーが正常に設定された後、図 16 に示すように EZ-Click ツール内の **CapSense output** タブをクリックして CapSense センサー データを表示します。

図 16. EZ-Click の CapSense Output タブ



5.3.3 ステップ C: raw カウントとベースラインの表示

1. 図 16 に示すように、raw カウントとベースラインを表示するには、**Select view** パラメーター内の **Button Output** オプションを選択します。
2. **Button** パラメーターでは、raw カウントとベースラインのデータを表示するセンサーを指定します。
3. **Graph** パラメーターでは、「**Raw count vs Baseline**」オプションを選択します。その他のすべてのパラメーターを変更せず初期値のままにします。

   これらパラメーターの詳細は、EZ-Click ユーザー ガイドを参照してください。
4. **Start** ボタンをクリックし、**Button** パラメーターで指定したセンサーの raw カウントとベースラインを取得し表示します。

   すべてのセンサーの raw カウント値を同時に表示するには、**Select view** パラメーターの **Parameter output** オプションと **Parameter** メニュー内の **Raw count** オプションを選択して **Start** ボタンをクリックします。

5.3.4 ステップ D: 差分カウントの表示

センサーの差分カウントの値は、「Automatic threshold」パラメータ (**CapSense sensor configuration** タブ内にある) が有効か無効であるかに応じて異なったウィンドウに表示されます。

「Automatic threshold」パラメーターが有効になる場合、以下の手順に従って差分カウントを表示します。

1. **Select view** パラメーターで、「**Button Output**」オプションを選択します。
2. **Graph** パラメーターでは、図 17 に示すように「 **Diff count vs Finger threshold**」オプションを選択し、「**Start**」ボタンをクリックしてセンサーの差分カウントを表示します。

「Automatic threshold」パラメーターが無効になる場合、以下の手順に従って差分カウントを表示します。

1. 「**Select view**」パラメーターで、「**Parameter output**」オプションを選択します。
2. **Parameter** メニューでは、図 18 に示すように「 **Difference count**」オプションを選択し、「**Start**」ボタンをクリックしてセンサーの差分カウントを表示します。

図 17.「Automatic Threshold」が有効時の差分カウントの表示

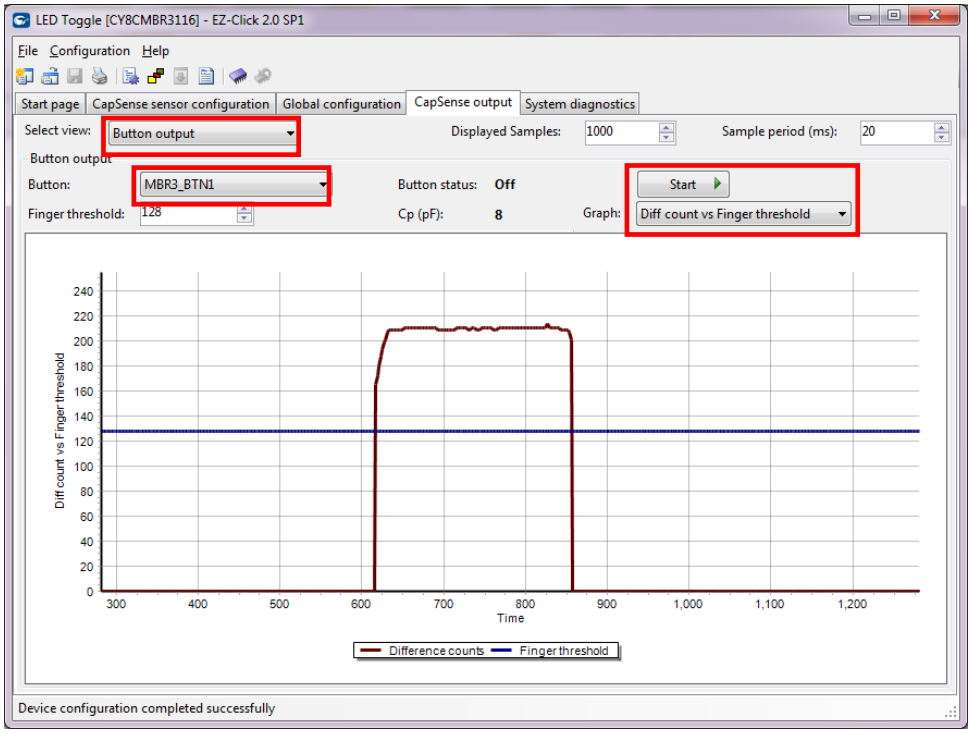
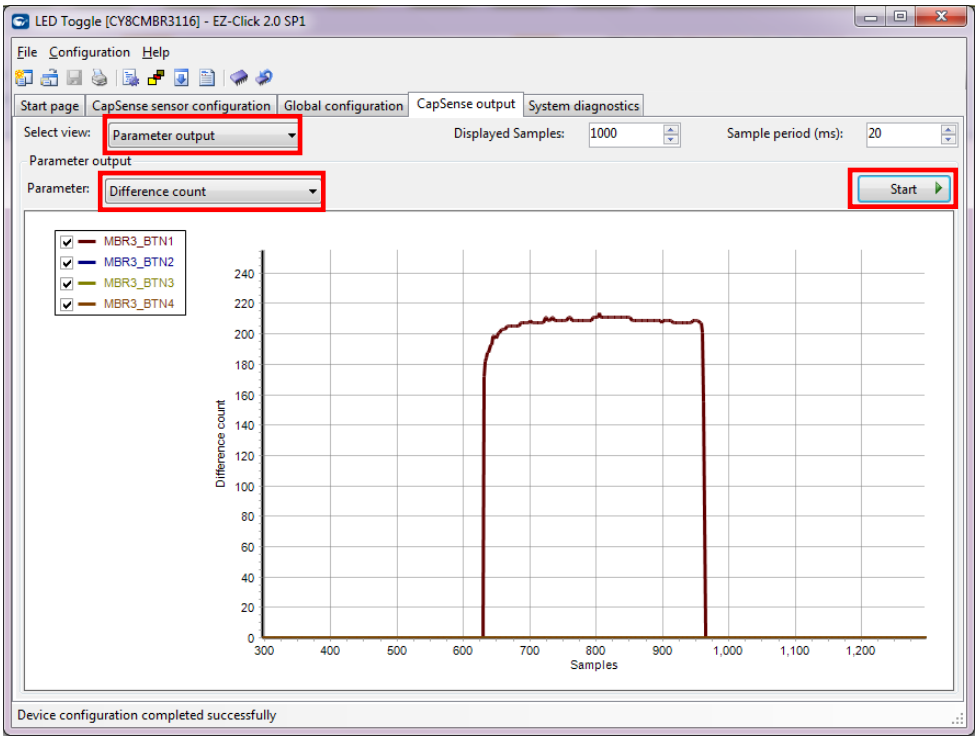


図 18.「Automatic Threshold」が無効時の差分カウントの表示



5.4 CY8CMBR20xx コントローラー向けの BCP 付き UART

5.4.1 システム要件

この方式を適用する際は以下のシステム コンフィギュレーションが必要です。

- 空いているシリアル通信 (COM／RS232) ポート[a]
- Microsoft Windows 9x、2000、XP、またはそれ以降のバージョン

5.4.2 サポートされるボーレート

ユーザー定義のボーレートで RS232 データを送信するようプログラムできるコントローラーと違い、CY8CMBR20xx デバイスのボーレートは不変のものです。表 33 では、CY8CMBR20xx デバイスとサポートされるボーレートのマッピングを示します。

表 3. CY8CMBR20xx がサポートするボーレート

CY8CMBR20xx	ボーレート
CY8CMBR2044	117.6kbps
CY8CMBR2010／CY8CMBR2016	115.2kbps

UART-USB ブリッジを有効にして RS232 データ (117.6kbps および 115.2kbps で出力) を読み出す方法については、「CY3240-I2USB ブリッジによる UART-USB ブリッジの有効化」を参照してください。

この方式で CapSense データを表示するために、「ステップ A: BCP のインストール」および「ステップ D: BCP による CapSense センサー データの読み出し」に従ってください。

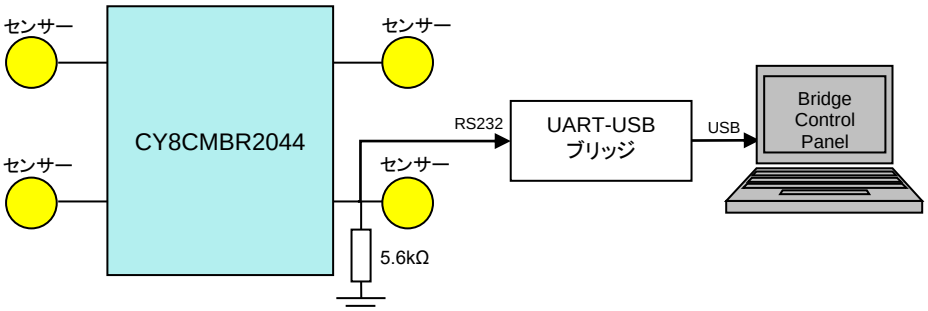
CY8CMBR20xx デバイスはピンにデバッグ データを送信するよう設定できます。シリアル デバッグ データの形式については、対応するデバイスのデータシートを参照してください。raw カウント、ベースライン、信号 (差分カウント) に加えて、CY8CMBR20xx デバイスは以下の情報も送信します。

- ファームウェア レビジョン
- CapSense ボタン タッチ ステータス
- GPO 出力ステータス
- 各 CapSense センサーの寄生容量

5.4.3 CY8CMBR2044 からのデータ読み出し

CY8CMBR2044 は、センサー ピンから CapSense データ (RS232 形式) を (117.6kbps のボーレートで) 送信します。センサー ピンは、5.6kΩ の抵抗でグランドにプルダウンされています。抵抗の許容誤差については、デバイス データシートを参照してください。図 19 に、CapSense データを送信するよう CY8CMBR2044 を設定する方法を示します。

図 19.CY8CMBR2044 ハードウェア セットアップ



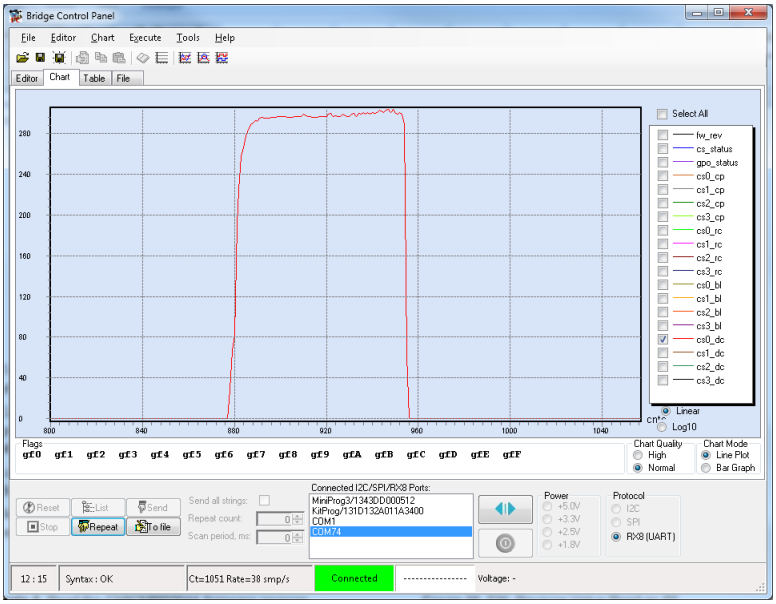
1. CY8CMBR2044 デバイスからデータを読み出すために、RS232 データを 117.6kbps のボーレートで読み出すようプログラムされた UART-USB ブリッジを使用します。詳細は、「CY3240-I2USB ブリッジによる UART-USB ブリッジの有効化」を参照してください。

[a] UART-USB ブリッジがシリアル (RS232) データを読み出すために使用される場合、この要件は必須ではありません。

2. グランドにプルダウンされている CY8CMBR2044 センサー ピンを CY3240-I2USB ブリッジの I2C_SDA ピンに接続します。CY8CMBR2044 の GND 端子を CY3240-I2USB ブリッジの GND ピンに接続します。CY3240-I2USB のピン配置の詳細は、CY3240-I2USB ブリッジ ユーザー ガイドを参照してください。
3. 「ステップ D: BCP による CapSense センサー データの読み出し」を参照し、UART データの監視のために Bridge Control Panel を設定します。
4. 本アプリケーション ノートとともに提供されたコマンド ファイル *CY8CMBR2044.iic* および変数設定ファイル *CY8CMBR2044.ini* を Bridge Control Panel にロードします。
5. コマンド全体を選択し、**Repeat** ボタンをクリックしてデータを読み出します。次に、Chart タブに進んで必要な変数を監視します。図 20 に、CS0 の差分カウントの監視を示します。

   センサー ステータスの各ビットおよび GPO ステータスの変数はそれぞれ対応するセンサーまたは GPO の状態を示すことに注意してください。例えば、センサー ステータス 0x08 は CS3 で指の接触が検知されることを示します。同様に、GPO ステータス 0x04 は GPO2 がアサート (アクティブ LOW) されることを示します。

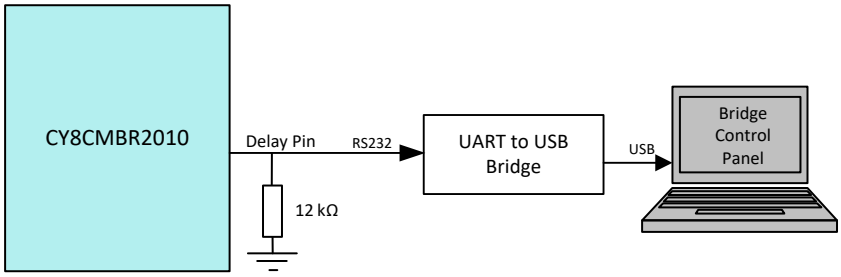
図 20. CS0 の差分カウントの監視 (CY8CMBR2044 の場合)



5.4.4 CY8CMBR2010 からのデータ読み出し

CY8CMBR2010 は、Delay ピンから CapSense データ (RS232 形式) を (115.2kbps のボーレートで) 送信します。Delay ピンは 12kΩ 抵抗でグランドにプルダウンされています。抵抗の許容誤差については、デバイス データシートを参照してください。図 21 に、CapSense データを送信するよう CY8CMBR2010 を設定する方法を示します。

図 21. CY8CMBR2010 ハードウェア セットアップ

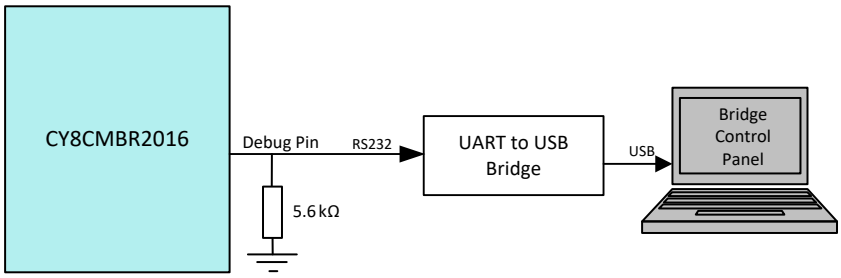


1. CY8CMBR2010 デバイスからデータを読み出すには、RS232 データを 115.2kbps のボーレートで読み出すようプログラムされた UART-USB ブリッジを使用します。詳細は、「CY3240-I2USB ブリッジによる UART-USB ブリッジの有効化」を参照してください。
2. CY8CMBR2010 の Delay ピンを CY3240-I2USB ブリッジの I2C_SDA ピンに接続します。CY8CMBR2010 の GND 端子を CY3240-I2USB ブリッジの GND ピンに接続します。CY3240-I2USB のピン配置の詳細は、CY3240-I2USB ブリッジ ユーザー ガイドを参照してください。
3. 「ステップ D: BCP による CapSense センサー データの読み出し」を参照し、UART データの監視のために Bridge Control Panel を設定します。
4. 本アプリケーション ノートとともに提供されたコマンド ファイル *CY8CMBR2010_xxxx.iic* と変数設定ファイル *CY8CMBR2010_xxxx.ini* を Bridge Control Panel にロードします。Bridge Control Panel が最大 32 の名付けられた変数しか監視できないため、Bridge Control Panel で CY8CMBR2010 が送信したデータのすべてを同時に監視することはできません。しかし、コマンド ウィンドウから任意のバイト数を読み出せます。したがって、選択的にデータを監視するために、複数のコマンドおよび変数設定用ファイルがこのデバイスに組み込まれています。以下はこのデバイスに提供されたコマンド ファイルの一覧です。各コマンド ファイルは対応する変数設定ファイルがあります。
   - *CY8CMBR2010_SensorData.iic*: すべての 10 個のボタンのセンサー状態、GPO 状態、CapSense スキャン データ (raw カウント、ベースライン、差分カウント) を監視します。
   - *CY8CMBR2010_DiagData.iic*: ファームウェア レビジョン、Cp、SNR、システム診断の結果などの他のデータを監視します。
5. コマンド全体を選択し、**Repeat** ボタンをクリックしてデータを読み出します。次に、Chart タブに進んで必要な変数を監視します。

5.4.5 CY8CMBR2016 からのデータ読み出し

CY8CMBR2016 は、CapSense データ (RS232 形式) を (115.2kbps のボーレートで) Debug ピンから送信します。「Debug」ピンは、5.6kΩ 抵抗でグランドにプルダウンされています。抵抗の許容誤差については、デバイス データシートを参照してください。図 22 に、CapSense データを送信するために CY8CMBR2016 を設定する方法を示します。

図 22.CY8CMBR2016 ハードウェア セットアップ



注:

1. CY8CMBR2016 デバイスからデータを読み出すには、RS232 データを 115.2kbps のボーレートで読み出すようプログラムされた UART-USB ブリッジを使用します。詳細は、「CY3240-I2USB ブリッジによる UART-USB ブリッジの有効化」を参照してください。
2. CY8CMBR2016 のデバッグ ピンを CY3240-I2USB ブリッジの I2C_SDA ピンに接続します。CY8CMBR2016 の GND 端子を CY3240-I2USB ブリッジの GND ピンに接続します。CY3240-I2USB のピン配置の詳細は、CY3240-I2USB ブリッジ ユーザー ガイドを参照してください。
3. 「ステップ D: BCP による CapSense センサー データの読み出し」を参照し、UART データの監視のために Bridge Control Panel を設定します。

4. 本アプリケーション ノートとともに提供されたコマンド ファイル *CY8CMBR2016_xxxx.iic* と変数設定ファイル *CY8CMBR2016_xxxx.ini* を Bridge Control Panel にロードします。Bridge Control Panel が最大 32 の名付けられた変数しか監視できないため、Bridge Control Panel で CY8CMBR2016 が送信したデータのすべてを同時に監視することはできません。しかし、コマンド ウィンドウから任意のバイト数を読み出せます。したがって、選択的にデータを監視するために、複数のコマンドおよび変数設定用ファイルがこのデバイスに組み込まれています。以下はこのデバイスに提供されたコマンド ファイルの一覧です。各コマンド ファイルは対応する変数設定ファイルがあります。
   - *CY8CMBR2016_SensorData_CS0toCS9.iic*: センサーCS0～CS9 のセンサー状態、CapSense スキャン データ (raw カウント、ベースライン、差分カウント) を監視します。
   - *CY8CMBR2016_SensorData_CS10toCS15.iic*: センサーCS10～CS15 のセンサー状態、CapSense スキャン データ (raw カウント、ベースライン、差分カウント) を監視します。
   - *CY8CMBR2016_CpData.iic*: ファームウェア レビジョンとすべてのセンサーの CP を監視します。
5. コマンド全体を選択し、**Repeat** ボタンをクリックしてデータを読み出します。次に、Chart タブに進んで必要な変数を監視します。

6 CY3240-I2USB ブリッジによる UART-USB ブリッジの有効化

本節は次の 2 つのトピックに触れています。

- 117.6kbps と 115.2kbps でクロックアウトされた RS232 データを読み出すために CY3240-I2USB ブリッジ ハードウェアを UART-USB ブリッジとして機能するよう構成する方法
- UART-USB ドライバーをインストールするための段階を追った手順

6.1 CY3240-I2USB から UART-USB ブリッジの構成

USB_UART_Bridge.hex ファイルで CY3240-I2USB ブリッジ ハードウェアをプログラムします。

- 117.6kbps で出力される RS232 データを読み出す場合、このファイルを *BaudRate117.6_UART_TO_USB.zip* ファイルから入手します。
- 115.2kbps で出力される RS232 データを読み出す場合、このファイルを *BaudRate115.2_UART_TO_USB.zip* ファイルから入手します。

ブリッジをプログラムする方法については、CY3240-I2USB ブリッジ ユーザー ガイドの「3.2.1. USB-I2C ブリッジのプログラミング」セクションを参照してください。

6.2 UART-USB ドライバーのインストール

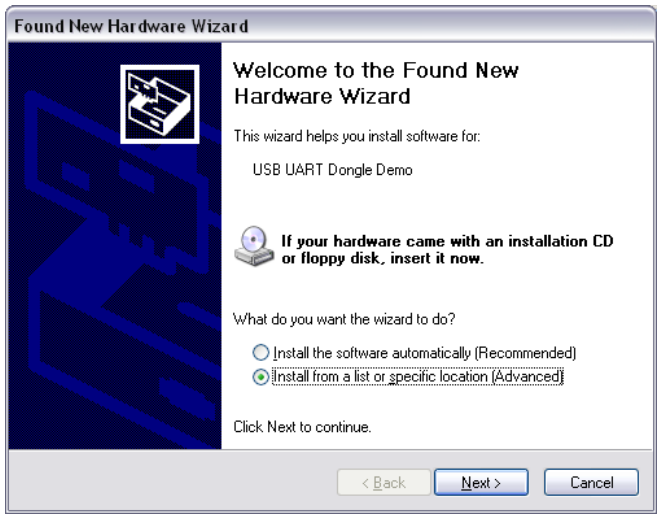
1. プログラミングを終了したら、CY3240-I2USB ブリッジ ハードウェアを PC USB ポートに接続します。デバイスが初めて PC に接続されると、新しいハードウェア ウィザードが起動します。
2. 図 23 に示すように、「**No, not this time**」(いいえ、今回は接続しない) を選択して、「**Next**」(次へ) をクリックします。

図 23. 新しいハードウェア ウィザード – ステップ 1



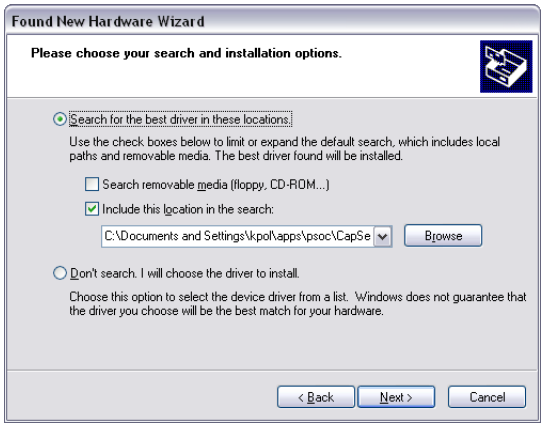
3. 図 24 に示すように、「**Install from a list or specific location (Advanced)**」(一覧または特定の場所からインストールする (上級)) を選択して「**Next**」(次へ) をクリックします。

図 24. 新しいハードウェア ウィザード – ステップ 2



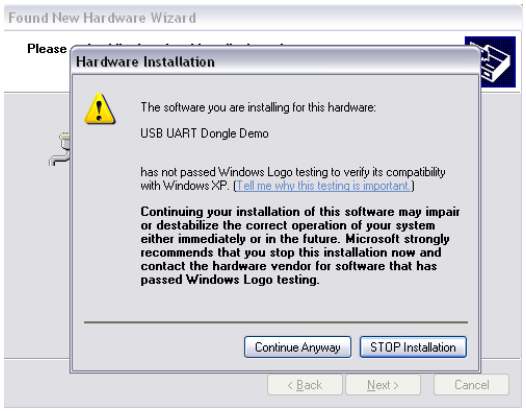
4. 「**Search for the best driver in these locations**」(次の場所で最適なドライバーを検索する) と「**Include this location in the search**」(次の場所を含める) を選択して、*UART_TO_USB.hex* ファイルを格納するフォルダを指すパスを設定します。図 25 に示すように、「**Next**」(次へ) をクリックします。

図 25. 新しいハードウェア ウィザード – ステップ 3



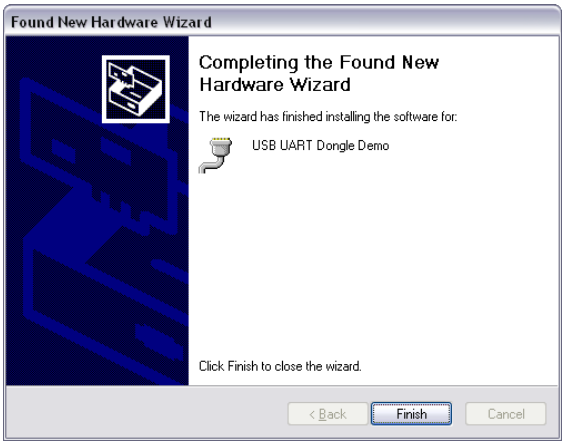
5. 図 26 に示すように、「**Continue Anyway**」(続行) をクリックします。

図 26. 新しいハードウェア ウィザード – ステップ 4



6. 図 27 に示すように、「**Finish**」(完了) をクリックしてドライバのインストールを完了します。

図 27. 新しいハードウェア ウィザード – ステップ 5



7 用語集

AMUXBUS

入出力ピンを複数の内部アナログ信号に接続する PSoC 内にあるアナログ マルチプレクサ バスです。

ベースライン

センサーに人の指で触らないときの raw カウントの傾向を推定しファームウェア アルゴリズムから得られる値です。ベースラインは raw カウントの突然の変化に敏感性が低く、差分カウントを計算するためのリファレンス点を提供します。

ボタン／ボタン ウィジェット

センターに対応しており、センサーのアクティブ状態または非アクティブ状態 (すなわち、2 つだけの状態) を報告するウィジェットです。例えば、センサー上の指の「タッチあり」または「タッチなし」の状態を検出できます。

差分カウント

raw カウントとベースラインの差分です。差分が負数であるか、またはノイズ閾値未満である場合、差分カウントは常に 0 に設定されます。

静電容量センサー

静電容量の変化によってタッチまたは近づいている物体に反応する導電体および基板 (プリント回路基板 (PCB) 上の銅ボタンなど) です。

CapSense®

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 ソリューションです。業界 2 位に対して、4 倍の販売実績がある業界 No.1 ソリューションです。

CapSense メカニカル ボタン リプレースメント (MBR)

メカニカル ボタンを静電容量ボタンにアップグレードするサイプレスの構成可能なソリューションであり、センサー パラメーターの設定に必要な設計工数を最小限に抑え、ファームウェアの開発も不要とします。これらのデバイスは CY8CMBR3XXX および CY8CMBR2XXX のファミリを含んでいます。

重心／重心位置

スライダー分解能の指定した範囲内のスライダー上の指の位置を示す数です。この数は CapSense 重心計算アルゴリズムにより算出されます。

補正 IDAC

過剰なセンサーC_Pを補正するために CSD により使用されるプログラム可能な定電流源です。この IDAC は、変調 IDAC と違って、CSD ブロックでシグマ デルタ変調器によって制御されません。

CSD

CapSense シグマ デルタ (CSD) は、静電容量センシングのアプリケーション用に自己容量を測定するサイプレスの特許取得済み方法です。

CSD モードでは、センシング システムは電極の自己容量を測定し、指の有無を識別するために自己容量の変化が検出されます。

デバウンス

有効なタッチとなるためにタッチがある必要な連続スキャン サンプル数を定義するパラメーターです。このパラメーターはスプリアス タッチ信号を除去するために役立ちます。

指のタッチは、差分カウントがスキャン サンプルの連続デバウンス数で [指閾値 + ヒステリシス] を超える場合にのみ報告されます。

被駆動シールド

シールド電極がセンサー スイッチング信号と同じ位相および振幅を持つ信号によって駆動され、耐液性を有効にするために CSD によって使用される技術です。

電極

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

Finger Threshold (指閾値)

センサーの状態を確定するためにヒステリシスと一緒に使用されるパラメーターです。センサーの状態は、差分カウントが [指閾値 + ヒステリシス] を上回る場合、オンとして報告され、差分カウントが [指閾値 – ヒステリシス] を下回る場合、オフとして報告されます。

連動センサー

複数のセンサーを連動して、単一のセンサーとしてスキャンする方法です。近接センシング用のセンサーの領域を増やし、消費電力を減少させるために用いられます。

システムが低電力モードにある場合の消費電力を削減するために、センサーは個別にスキャンされず、これらのすべてを連動して単一のセンサーとしてスキャンされ、時間を短縮させます。ユーザーがセンサーに触れる場合、システムはアクティブモードに移行し、すべてのセンサーを個別にスキャンしてアクティブになったセンサーを検出します。

PSoC はファームウェアでセンサー連動をサポートし、これにより、複数のセンサーはスキャンのために AMUXBUS と同時に接続できます。

ジェスチャー

ジェスチャーはスワイプやピンチズームなどのユーザーの行動です。CapSense は事前に定義されたタッチ パターンに基づいて異なるジェスチャーを識別するジェスチャー検出機能を備えています。CapSense コンポーネントでは、ジェスチャー機能はタッチパッド ウィジェットのみによってサポートされます。

ガード センサー

ボタン センサーと同様に PCB 上のすべてのセンサーを取り囲み、液体流を検出するために使用される銅配線です。ガードセンサーがトリガーされると、ファームウェアは誤ったタッチを防ぐために、すべての他のセンサーのスキャンを無効にできます。

ハッチ フィル／ハッチ グランド／ハッチド グランド

静電容量センシングの PCB を設計する際、良好なノイズ耐性のために接地した銅面をセンサーの周りに配置する必要があります。しかし、ベタ グランドはセンサーの期待されない寄生静電容量を増加させます。そのため、グランドは特別なハッチパターンで充填する必要があります。ハッチ パターンはメッシュのように密接に配置され、交差されるラインがあります。充填率はラインの幅および 2 本のライン間の間隔によって決まります。耐液性の場合、シールド電極 として呼ばれるこのハッチ フィルはグランドの代わりにシールド信号で駆動されます。

ヒステリシス

システム ノイズに起因してセンサー状態の出力がランダムにトグルすることを回避し、センサーの状態を決定するために指閾値と一緒に使用されるパラメーターです。「Finger **Threshold (指閾値)**」を参照してください。

IDAC (電流出力デジタル-アナログ変換器)

CapSense および ADC 動作用の PSoC 内のプログラマブルな定電流源です。

耐液性

水滴、液体流や霧が存在する環境でも確実に動作する静電容量センシング システムの能力です。

リニア スライダー

指の物理的な位置 (単一の軸で) を検出するために特定の直線状で配置された複数のセンサーを含むウィジェットです。

低ベースライン リセット

raw カウントが負のノイズ閾値を異常的に下回るスキャン サンプルの最大数を表すパラメーターです。低ベースライン リセットの値を超える場合、ベースラインは現時点の raw カウントにリセットされます。

手動チューニング

CapSense パラメーターの手動設定 (または手動チューニング) プロセスです。

マトリックス ボタン

マトリックス状で配置された 2 つ以上のセンサーから成り、垂直方向および水平方向に配置されるセンサーの交差部に人の指 (タッチ) の有無を検出するために使用されるウィジェットです。

水平軸上のセンサーの数を M、垂直軸上のセンサーの数を N とした場合、マトリックス ボタン ウィジェットは M + N 本のポート ピンだけを使用して、合計で M x N 個の交差部を監視できます。

CSD センシング方式 (自己容量) を使用する場合、このウィジェットは同時に 1 点のみの交差位置で有効なタッチを検出できます。

変調コンデンサ (CMOD)

自己容量センシング モードでの CSD ブロックの動作のために必要な外部コンデンサです。

変調器クロック

センサーのスキャン間に CSD ブロックから変調器出力をサンプリングするために使用されるクロック ソースです。このクロックが raw カウント カウンターにも供給されます。スキャン時間 (事前および事後処理時間を除く) は [$(2^N - 1)$/変調器クロック周波数] (N はスキャンの分解能) により計算されます 。

変調 IDAC

変調 IDAC はプログラマブルな定電流源であり、この出力は V_{REF} の AMUXBUS 電圧を維持するために、CSD ブロックのシグマ デルタ変調器の出力によって制御 (オン／オフ) されます。この IDAC によって供給される平均電流は、センサー コンデンサが引き出した平均電流に等しいです。

相互容量

ある電極 (例えば TX) と他の電極 (例えば RX) 間の静電容量は相互容量として知られています。

負のノイズ閾値

負の方向に出るスプリアス信号から通常のノイズを識別するために使用される閾値です。このパラメーターは、低ベースライン リセット パラメーターとともに使用されます。

raw カウントが負のノイズ閾値を超えない (すなわち、ベースラインと raw カウントの差 (ベースライン – raw カウント) が負のノイズ閾値未満である) 限り、ベースラインは raw カウントの変化を追跡するために更新されます。

負の方向でスプリアス信号をトリガーする可能性があるシナリオは次のとおりです。電源投入時にセンサーに指が触れるとき、センサーの近く配置される金属の物体を除去するとき、耐液性のある CapSense 製品の水分を除去するとき、および他の急激な環境変化があるときです。

ノイズ (CapSense ノイズ)

センサーがオフ状態 (タッチなし) のときにピーク ツー ピーク カウントとして測定される raw カウントの変化です。

ノイズ閾値

センサー用にノイズから信号を識別するために使用されるパラメーターです。[raw カウント – ベースライン] の差分がノイズ閾値を超える場合、おそらく有効な信号を示します。差分がノイズ閾値に下回る場合、raw カウントはノイズのみを含みます。

オーバーレイ

静電容量センサーをカバーしタッチ面として機能するプラスチックやガラスなどの非導電材料です。センサーがある PCB はオーバーレイの下に直接配置されるか、またはスプリングを介して接続されます。製品の筐体は常にオーバーレイです。

寄生容量 (C_P)

寄生容量は PCB 配線、センサパッド、ビアおよびエア ギャップによって与えられるセンサー電極の固有容量です。寄生容量は CSD の感度を減らすため、望ましくないものです。

近接センサー

あらゆる物理的な接触なしに近くの物体の存在を検知できるセンサーです。

ラジアル スライダー

指の物理的な位置を検出するために特定の円形状に配置された複数のセンサーを含むウィジェットです。

raw カウント

センサーの物理的静電容量を示す CapSense ハードウェア ブロックの未処理デジタル カウント出力です。

リフレッシュ間隔

センサーの 2 回の連続スキャンの間の時間です。

スキャン分解能

CSD ブロックによって生成される raw カウントの分解能 (ビット数) です。

スキャン時間

センサーのスキャンを完了するのに要する時間です。

自己容量

回路のグランドと電極間の静電容量です。

感度

センサー静電容量の変化に応じる raw カウントの変化であり、カウント/pF の単位で表します。センサーの感度は基板レイアウト、オーバーレイ特性、センシング方式およびチューニング パラメーターに依存します。

センス クロック

CSD センシング方式用のスイッチト キャパシタ回路のフロント エンドを実装するために使用されるクロック ソースです。

センサー

「静電容量センサー」を参照してください。

センサー自動リセット

システム故障の際、または金属物体がセンサーの近くに連続的に存在する際に、センサーが誤ったタッチ状態を無期限に報告してしまうことを防ぐための設定です。

センサー自動リセット機能が有効になった場合、ベースラインは差分カウントがノイズ閾値を超えても常に更新されます。このように、センサーが無期限のオン状態を報告しないようにします。センサー自動リセットが無効なとき、ベースラインは差分カウントがノイズ閾値を下回った場合にのみ更新されます。

センサー連動

「連動センサー」を参照してください。

シールド電極

センサーの周囲を覆う銅トレースで水または他の液体による誤タッチを防止します。シールド電極は CSD ブロックからのシールド信号出力によって駆動されます。「被駆動シールド」を参照してください。

シールド タンク コンデンサ (C_{SH})

高い寄生容量を持つ広いシールド層がある場合、CSD シールドの駆動能力を強化するために使用されるオプションの外部コンデンサ (C_{SH} タンク コンデンサ) です。

信号 (CapSense 信号)

差分カウントは信号とも呼ばれます。「差分カウント」を参照してください。

信号対ノイズ比 (SNR)

タッチしたときのセンサーの信号とタッチしないときのセンサーのノイズ信号との比率です。

スライダー分解能

スライダーが分解された指の位置の総数を示すパラメーターです。

SmartSense™ 自動チューニング

設計フェーズの後で最適性能のために、センシング パラメーターを自動的にセットし、システム、製造および環境変化に対し連続的に補正する CapSense アルゴリズムです。

タッチパッド

特定の水平と垂直な様式で配置された複数のセンサーから成り、タッチの X および Y 位置を検出するウィジェットです。

トラックパッド

「タッチパッド」を参照してください。

チューニング

CapSense の動作に必要なさまざまなハードウェアおよびソフトウェアまたは閾値パラメーターの最適値を決定するプロセスです。

V_{REF}

PSoC 内にあるプログラマブル電圧リファレンス ブロックであり、CapSense および ADC の動作に使用されます。

ウィジェット

単一センサーまたは同様のセンサー グループで構成される CapSense コンポーネントのユーザー インターフェース要素です。ボタン、近接センサー、リニア スライダー、ラジアル スライダー、マトリックス ボタンおよびタッチパッドはサポートされたウィジェットです。

8 まとめ

本アプリケーション ノートは、I²C または UART インターフェースを使用して、サイプレスの CapSense コントローラーから CapSense デバッグ データを監視する方法を解説しました。また、Bridge Control Panel と EZ-Click ツールを使用して CapSense データを監視する方法についても詳しく説明しました。

著者について

氏名: Kurian Polachan

役職: シニア アプリケーション エンジニア

改訂履歴

文書名: AN2397 - PSoC® 1 および CapSense® コントローラー – CapSense データ監視ツール

文書番号: 001-79063

版	ECN	発行日	変更内容
**	3611341	05/10/2012	これは英語版 001-41446 Rev *C を翻訳した日本語版 001-79063 Rev. **です
*A	4928458	09/25/2015	これは英語版 001-41446 Rev. *I を翻訳した日本語版 001-79063 Rev. *A です。
*B	5830264	07/24/2017	ロゴと著作権を更新しました。
*C	6655978	01/08/2020	これは英語版 001-41446 Rev. *L を翻訳した日本語版 001-79063 Rev. *C です。

ワールドワイドな販売と設計サポート

サイプレスは、事業所、ソリューションセンター、メーカー代理店、および販売代理店の世界的なネットワークを保持しています。お客様の最寄りのオフィスについては、サイプレスのロケーション ページをご覧ください。

製品

Arm® Cortex® Microcontrollers	cypress.com/arm
車載用	cypress.com/automotive
クロック&バッファ	cypress.com/clocks
インターフェース	cypress.com/interface
IoT (モノのインターネット)	cypress.com/iot
メモリ	cypress.com/memory
マイクロコントローラ	cypress.com/mcu
PSoC	cypress.com/psoc
電源用 IC	cypress.com/pmic
タッチセンシング	cypress.com/touch
USB コントローラー	cypress.com/usb
ワイヤレス	cypress.com/wireless

PSoC®ソリューション

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

サイプレス開発者コミュニティ

コミュニティ | Projects | ビデオ | ブログ | トレーニング | Components

テクニカル サポート

cypress.com/support

本書で言及するその他すべての商標または登録商標は、それぞれの所有者に帰属します。

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2007-2020. 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。