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。

www.infineon.com

PSoC® 1 入門

著者: Robert Murphy、Ashutosh Srivastava

関連プロジェクト: あり

関連部品ファミリ: すべての PSoC 1 ファミリ

ソフトウェア バージョン: PSoC Designer™ 5.4 SP1 およびそれ以降

関連アプリケーション ノート: アプリケーション ノートの完全なリストについては、ここをクリックしてください。

本アプリケーション ノートの最新版または関連プロジェクト ファイルについては、ここをご覧ください。

本アプリケーション ノート (AN75320) は、カスタム機能を実装できるプログラマブル デジタルとアナログ ブロックを内蔵した 8 ビット プロセッサである PSoC® 1 を紹介します。本アプリケーション ノートは PSoC 1 アーキテクチャ、開発ツールおよび初めての設計を作成する方法について説明します。本書はまた、PSoC 1 を迅速に深く理解できるように更なるリソースへのアクセスを提供します。

目次

1 はじめに 2
2 概要 2
2.1 PSoC Designer 3
2.2 サンプル コード 4
2.3 PSoC Designer Help 5
2.4 テクニカル サポート 5
3 PSoC 1 ファミリの比較 6
4 PSoC 1 の機能セット 7
4.1 M8C プロセッサおよびメモリ 8
4.2 プログラマブル デジタル サブシステム 8
4.3 プログラマブル アナログ サブシステム 8
4.4 システム全般リソース 8
4.5 GPIO システム 9
4.6 CapSense 9
4.7 ダイナミックなリコンフィギュアビリティ 10
5 開発ツール 10
5.1 ソフトウェア: PSoC Designer IDE 10
5.2 ハードウェア 11
6 PSoC 1 学習資料 11
6.1 PSoC 1 データシート 11
6.2 PSoC 1 Designer の学習 12
6.3 アプリケーション ノート 12
6.4 知識ベース記事 12
6.5 テクニカル リファレンス マニュアル (TRM) 12
6.6 デバイス エラッタ 12
6.7 テクニカル サポート 12
7 はじめての PSoC 1 設計 13
7.1 設計について 13
7.2 はじめての PSoC 1 設計の作成 13
7.3 PWM 信号のルーティング 18
7.4 コードの追加 19
7.5 ピン配置のコンフィギュレーション 21
7.6 ビルドおよびプログラム 21
7.7 CY3210-PSoCEval1 基板の設定 23
7.8 CY8CKIT-001 ボードの設定 25
8 まとめ 27
9 関連アプリケーション ノート 27

1 はじめに

PSoC 1 は、単一のチップ上にカスタム アナログとデジタル ペリフェラル機能、メモリおよび M8C マイクロコントローラーを集積した、真の**プログラマブルな組み込みシステムオンチップ**です。

これは、マイクロコントローラー (MCU) と外部アナログ／デジタル ペリフェラルの組み合わせを使用するアナログ／デジタル混在の組み込みシステムとは異なります。通常、このシステム タイプは MCU に加えて、オペアンプ、ADC、デジタル ASIC など多くの集積回路を必要とします。

PSoC 1 は、MCU と外部 IC の組み合わせの代わりに低コストのソリューションを提供しています。システム全体のコストを削減する以外、プログラマブル アナログとデジタル サブシステムにより、設計の高い柔軟性、インフィールド チューニングおよび短い製品化までの時間が可能になります。

PSoC 1 のシステム リソースは、全く異なる機能を実行するよう実行中にダイナミックにリコンフィギュレーションすることができます。ダイナミックなリコンフィギュレーションは、リソースが設計に消費された場合にも可能です。これにより、PSoC 1 のシステム リソースを再利用し、集積能力を最大限に発揮させることができます。

PSoC 1 の CapSense®という静電容量タッチ センシング機能は、比類のない信号対ノイズ比 (SNR) 、クラス最高の耐水性、および多種多様なセンサー タイプ (ボタン、スライダー、トラック パッド、近接センサーなど) を提供しています。

本アプリケーション ノートは、PSoC 1 のアーキテクチャおよび開発ツールを研究するのに役立ち、PSoC 1 の開発ツールである PSoC Designer™を使用して初めてのプロジェクトを作成する方法を案内します。本書はまた、PSoC 1 を迅速に深く理解できるように更なるリソースへのアクセスを提供します。

PSoC 1 に加えて、サイプレスの PSoC ポートフォリオは PSoC 3、PSoC 4 および PSoC 5LP デバイスを含んでいます。PSoC デバイスによってシステム アーキテクチャおよびペリフェラルが異なります。詳細については、「Cypress Platform PSoC Solutions Roadmap」を参照してください。

2 概要

サイプレスは、ユーザーが設計に適切な PSoC デバイスを選択し、デバイスを設計に迅速で効果的に統合するのに役立つ豊富な情報を www.cypress.com に掲載しています。リソースの総合リストについては、知識ベース記事「How to Design with PSoC® 1, PowerPSoC®, and PLC – KBA88292」を参照してください。以下は PSoC 1 のリソースの要約です。

- 概要: PSoC ポートフォリオ、PSoC ロードマップ
- 製品セレクタ: PSoC 1、PSoC 3、PSoC 4、PSoC 5LP
- さらに、PSoC Designer にはデバイス選択ツールが含まれています。
- アプリケーション ノート: サイプレスは、基本レベルから上級レベルまでの様々なトピックに触れる大量の PSoC アプリケーション ノートを提供します。以下は PSoC 1 入門用の推奨アプリケーション ノートです。
  - AN75320 - Getting Started with PSoC® 1
  - AN2094 - PSoC® 1 - Getting Started with GPIO
  - AN74170 - PSoC® 1 Analog Structure and Configuration with PSoC Designer™
  - AN2041 - Understanding PSoC® 1 Switched Capacitor Analog Blocks
  - AN2219 - PSoC® 1 Selecting Analog Ground and Reference

注: CY8C29X66 デバイスに関するアプリケーション ノートについては、ここをクリックしてください。

- 開発キット:
  - CY3210-PSoCEval1 は、CY8C25/26xxx デバイスを除いて、車載用デバイスを含むすべての PSoC 1 アナログ／デジタル混在信号アレイ ファミリに対応しています。このキットには、LCD モジュールやポテンショメーター、LED、ブレッドボードが含まれています。
  - CY3214-PSoCEvalUSB は、CY8C24x94 PSoC デバイスの開発用基板を備えています。この基板の特殊な特長には、USB および CapSense 開発とデバッグ サポートが含まれています。

注: CY8C29X66 デバイスに関する開発キットについては、ここをクリックしてください。

MiniProg1 および MiniProg3 デバイスは、フラッシュのプログラミングとデバッグ用のインターフェースを提供します。

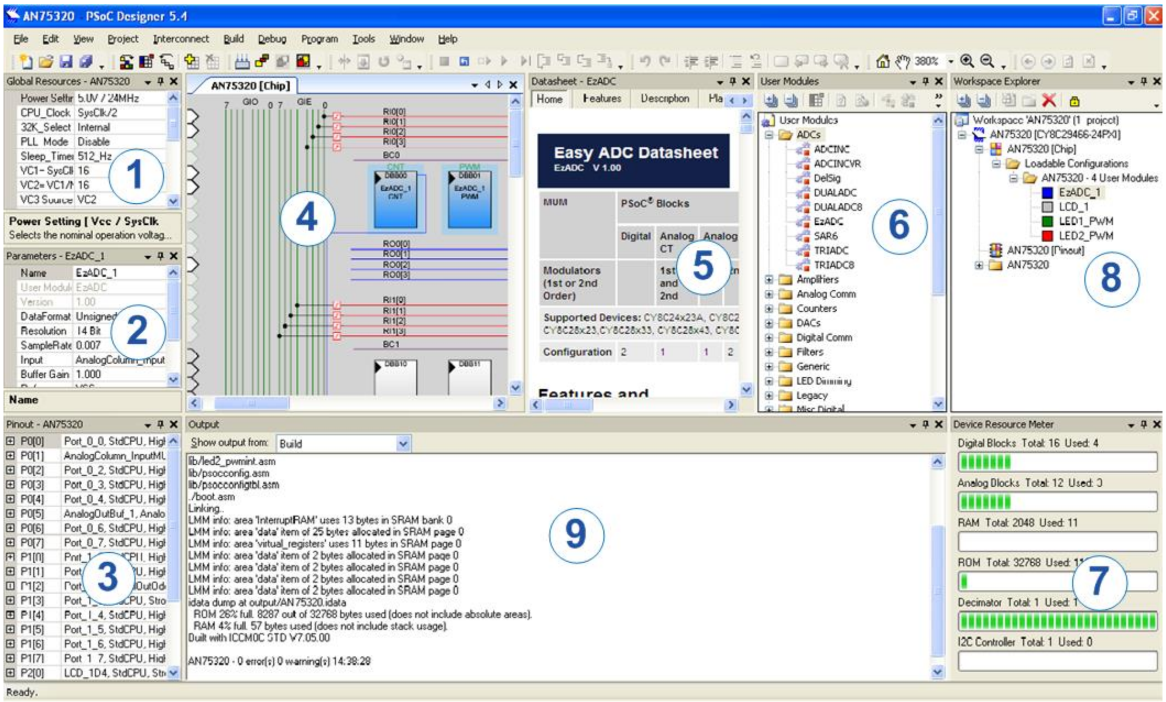
2.1 PSoC Designer

PSoC Designer は無料の Windows ベースの統合設計環境 (IDE) です。アプリケーション開発は、特性化済みのアナログとデジタル ペリフェラルのライブラリを使用してドラッグ アンド ドロップの設計環境で行われます。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可能です。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. 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. パラメーター: 選択しているユーザー モジュールのパラメーター
3. ピン配置: デバイスのピンに関する情報
4. チップ レベル エディター: 選択したチップで使用可能なリソースの図
5. データシート: 選択しているユーザー モジュールのデータシート
6. 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. デバイス リソース メーター: 現時点のプロジェクト コンフィギュレーション用のデバイス リソースの使用率
8. 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. 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 IDE の **Help** > **Documentation** をアクセスし、「Designer Specific Documents」フォルダを開いて「IDE User Guide.pdf」ドキュメントをご覧ください。

図 1. PSoC Designer のレイアウト



2.2 サンプル コード

以下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設計で始まり設計時間を短縮させることができ、PSoC Designer ユーザー モジュールが様々な用途にどのように利用できるかを示します。

http://www.cypress.com/documentation/code-examples/psoc-1-code-examples

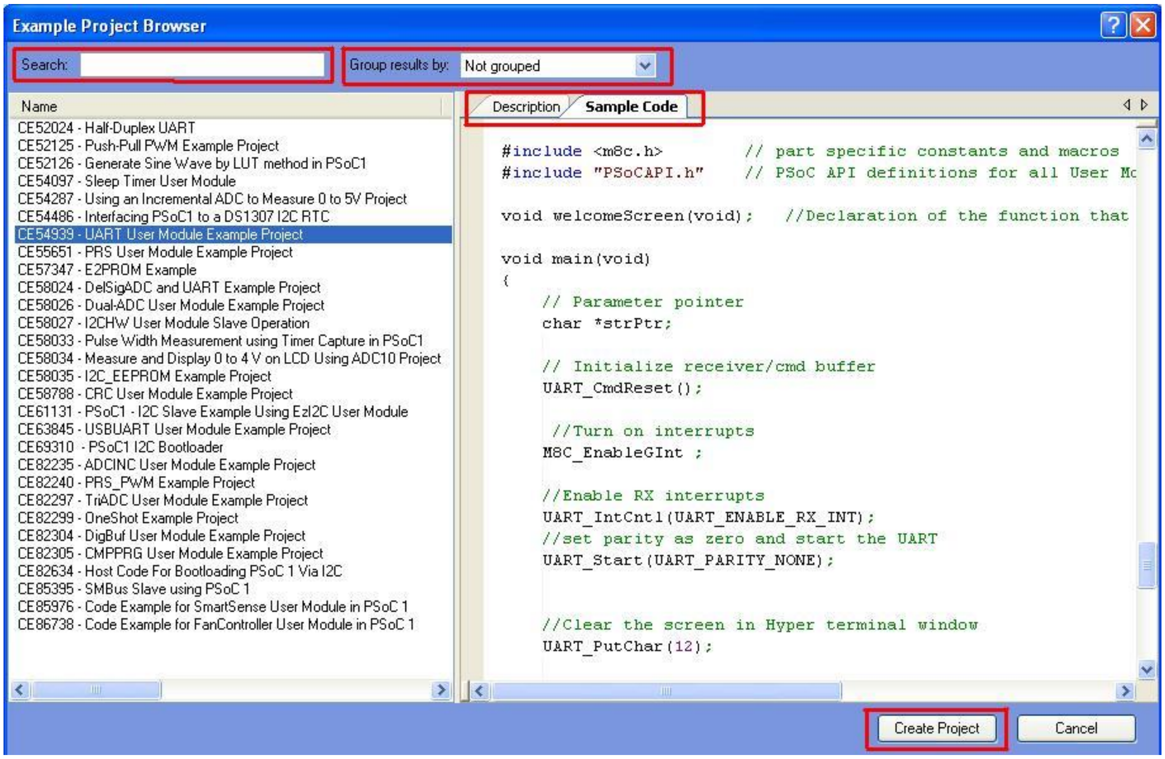
PSoC Designer に統合されているサンプル コードへアクセスするには、図 2 に示すように **Start Page > Design Catalog > Launch Example Browser** を順に選択してください。

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。

- プロジェクトをフィルタリングする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をレビューします (Description タブで)
- 選択したプロジェクトのサンプル コードをレビューします。このウィンドウからコードをプロジェクトにコピー アンド ペーストしてコード開発時間を短縮させることができます。
- さらに、選択に応じて新規プロジェクト (また、必要な場合は新規ワークスペース) を作成することができます。完成した基本的な設計で始まり設計時間を短縮させます。その設計をアプリケーションに適用することができます。

図 3. サンプル プロジェクトおよびサンプル コード



2.3 PSoC Designer Help

PSoC Designer ホームページへアクセスして PSoC Designer の最新版をダウンロードしてください。PSoC Designer を起動して次の項目に移動します。

- **IDE ユーザー ガイド**: **Help > Documentation > Designer Specific Documents > IDE User Guide.pdf** を選択します。このガイドは PSoC Creator プロジェクトを開発するための基礎知識を提供します。
- **シンプルなユーザー モジュールのサンプル コード**: **Start Page > Design Catalog > Launch Example Browser** を選択します。これらのサンプル コードは PSoC Designer ユーザー モジュールのコンフィギュレーションおよび使用方法をデモします。
- **テクニカル リファレンス マニュアル**: **Help > Documentation > Technical Reference Manuals** を選択します。このガイドは PSoC 1 デバイスのシステム機能を一覧にして説明します。
- **ユーザー モジュール データシート**: ユーザー モジュールを右クリックして「Datasheet」を選択します。このデータシートは選択されたユーザー モジュールのパラメーターおよび API を説明します。
- **デバイス データシート**: **Help > Documentation > Device Datasheets** を選択して特定の PSoC 1 デバイスのデータシートへアクセスします。
- **Imagecraft コンパイラ ガイド**: **Help > Documentation > Compiler and Programming Documents > C Language Compiler User Guide.pdf** を選択します。このガイドは Imagecraft コンパイラ固有の命令と関数について説明します。

2.4 テクニカル サポート

ご質問は、弊社のテクニカル サポート チーム対応いたします。サイプレスのテクニカル 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。

米国のお客様は、弊社フリーダイヤル (+1-800-541-4736 ext.8) までお電話いただければ、弊社のテクニカル サポート チームがご対応いたします。

早急な対応が求められる場合には、下記の方法をご利用ください。

- セルフ ヘルプ
- 所在地の販売代理店

3 PSoC 1 ファミリの比較

PSoC 1 には 13 デバイス ファミリがあります。表 1 は、これらのデバイス ファミリで対応されている機能を示します。

表 1. PSoC 1 デバイス セレクタの概要表

機能	デバイス ファミリ											
	29x66	28xxx	27x43	24x94	24x93	24x33	24x23A	23x33	22x45	21x45	21x34	21x23
フラッシュ (KB)	32K	16K	16K	16K	32K	8K	4K	8K	16K	8K	8K	4K
SRAM (KB)	2K	1K	256B	512B	2K	256B	256B	256B	1K	512B	512B	256B
ADC 1 (DS/SS)	14 ビット (DS)	4x14 ビット (DS)	11 ビット (DS)	14 ビット (DS)	10 ビット (INC)	11 ビット (DS)	11 ビット (DS)	11 ビット (DS)	8 ビット (SS)	8 ビット (SS)	10 ビット (SS)	10 ビット (SS)
ADC 2 (SAR)	-	10 ビット (150 Ksps)	-	-	-	8 ビット (300 Ksps)	-	8 ビット (300 Ksps)	10 ビット (150 Ksps)	10 ビット (150 Ksps)	-	-
コンパレータ	4	4	4	2	2	2	2	2	4	4	2	2
DAC (8 ビット)	4	4	4	2	-	2	2	2	-	-	-	-
PGA (x48 ゲイン)	4	4	4	2	-	2	2	2	-	-	-	-
TCPWM (16 ビット)	8	6	4	2	3 (タイマー)	2	2	2	4	2	2	2
UART/SPI	4	3	2	1	1 SPI	1	1	1	2	1	1	1
I2C	1	2	1	1	1 (スレーブ)	1	1	1	1	1	1	1
CapSense ボタン	-	43	-	49	-	-	-	-	37	23	23	-
GPIO	64	44	44	50	36	26	24	26	38	24	24	16
USB	-	-	-	FS 2.0	FS 2.0	-	-	-	-	-	-	-
ECO	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	N	N
SMP	Y	Y	Y	N	N	N	Y	N	Y	Y	Y	Y
MAC	2	2	1	2	0	1	1	1	1	1	0	0

FS = フル スピード、**SMP** = スイッチ モード ポンプ、**ECO** = 外部水晶振動子、**MAC** = 積和演算器、**INC** = インクリメンタル、**DS** = デュアル スロープ、**SS** = シングル スロープ、**PGA** = プログラマブル ゲイン アンプ、**TCPWM** = タイマー カウンター パルス幅変調器、**UART** = ユニバーサル非同期レシーバー トランスミッター、**SPI** = シリアル ペリフェラル インターフェース、**I2C** = インター インテグレーテッド サーキット

4 PSoC 1 の機能セット

PSoC 1 は図 4 に示すように、CPU コア、デジタル サブシステム、アナログ サブシステム、メモリ サブシステム、システム リソースを含む幅広い能力と機能を備えています。次の節は各機能について簡単に説明します。詳細については、PSoC 1 ファミリのデバイス データシート、テクニカル リファレンス マニュアル (TRM) および「PSoC 1 学習資料」にリストアップされたアプリケーション ノートを参照してください。

図 4. PSoC 1 アーキテクチャ (CY8C29466)

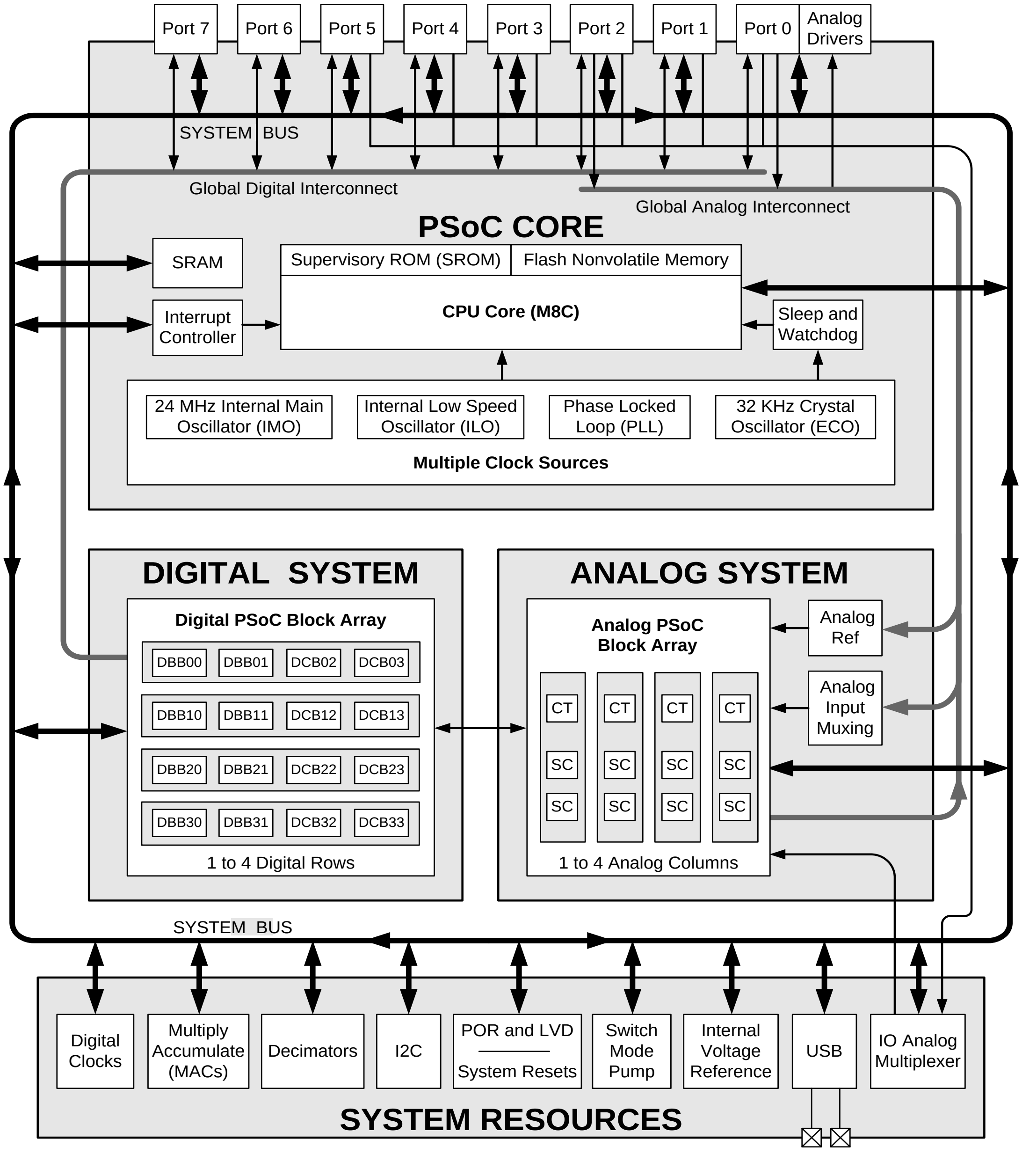


図 4 は CY8C29466 デバイスで対応されている機能を示します。これらの機能のサブセットは他のデバイスで使用可能です。表 1 を参照してください。

4.1 M8C プロセッサおよびメモリ

PSoC 1 は、ハーバード アーキテクチャの 8 ビット RISC CPU である M8C プロセッサを内蔵しています。これは 24MHz の最大周波数で動作可能で、4MIPS の性能を実現します。M8C には 37 の命令があります。表 1 に示すように、PSoC 1 デバイスは最大 2KB の SRAM と 32KB のフラッシュ メモリを内蔵します。詳細については、PSoC 1 テクニカル リファレンス マニュアルを参照してください。

4.2 プログラマブル デジタル サブシステム

PSoC のデジタル サブシステムは、そのプログラマビリティと配線システムによりユニークなものとなっています。表 1 は、各種の PSoC 1 製品および各デバイスで使用可能なデジタル リソースを一覧にします。PSoC 1 内のデジタル ブロックはデジタル基本ブロック (DBB) とデジタル通信ブロック (DCB) に分かれています。DBB 数と DCB 数は常に等しいです。

4.2.1 デジタル基本ブロック (DBB)

DBB は、タイマーやカウンター、PWM などとして機能させるようにプログラムできる基本的なコンフィギュレーション可能なデジタル リソースです。各 DBB には 8 ビット リソースを割り当てられます。16 ビット、24 ビット、または 32 ビット デジタル リソースを構築するにはそれぞれ 2 個、3 個、または 4 個の DBB が必要です。

4.2.2 デジタル通信ブロック (DCB)

DCB には、基本的なデジタル リソース (タイマーやカウンター、PWM など) を割り当てるだけでなく、SPI や UART などの通信リソースを割り当てることもできます。DBB コンポーネントを DCB に配置することは可能ですが、DCB リソースを DBB に配置することはできません。

4.3 プログラマブル アナログ サブシステム

PSoC 1 独自のアナログ サブシステムは、列に配置されたアナログ ブロックで構成されています。これらのアナログ ブロックは連続時間 (CT) ブロックとスイッチト キャパシタ (SC) ブロックに分かれています。

4.3.1 連続時間 (CT) ブロック

PSoC 内の CT ブロックは、コンパレーターまたはプログラマブル ゲイン アンプ (PGA) としてコンフィギュレーションできるプログラマブル アナログ ブロックです。CT ブロックは低ノイズ オペアンプと低オフセット オペアンプを中心に構築されています。CT ブロックには多くのアナログ マルチプレクサがあり、高いコンフィギュアビリティを実現します。CT ブロックの構造と構成の詳細については、PSoC 1 テクニカル リファレンス マニュアルの第 22 章を参照してください。

4.3.2 スイッチト キャパシタ (SC) ブロック

CT ブロックと同様に、SC ブロックはアナログ マルチプレクサに囲まれた低ノイズ オペアンプと低オフセット オペアンプを中心に構築されています。これらのブロックは、オペアンプとマルチプレクサの周囲のキャパシタとスイッチのグループによりユニークなものとなっています。CT ブロックとは異なり、SC ブロックに抵抗アレイはありません。SC ブロックのアーキテクチャの詳細については、PSoC 1 テクニカル リファレンス マニュアルの第 23 章または「AN2041 – Understanding PSoC 1 Switch Capacitor Analog Blocks」を参照してください。

4.4 システム全般リソース

4.4.1 クロッキング システム

PSoC は、複数のクロック ソースを持った高度なクロック システムを搭載しており、その中の多くはプログラム可能です。主要のクロック ソースは、内蔵の 24MHz 内部主振動子 (IMO) または 24MHz 以下の外部クロック ソースから派生できます。さらに、32kHz 振動子回路または内部低速振動子 (ILO) を低速振動子に使用することができます。PSoC 1 で使用可能なクロックの詳細については、「AN32200 - PSoC® 1 - Clocks and Global Resources」を参照してください。

4.4.2 スイッチ モード ポンプ (SMP)

スイッチ モード ポンプ (SMP) は PSoC でサポートされている DC/DC ブースト回路であり、デバイスが単一の 1.5V バッテリで動作できるようにします。PSoC には、ブースト回路の実行用に内蔵 FET および独立した PWM ハードウェアがあります。ブースト回路の構成要素は外部バッテリ、インダクタ、ダイオード、コンデンサ各 1 個のみです。詳細については、PSoC 1 テクニカル リファレンス マニュアルの第 30 章を参照してください。

4.4.3 積和演算器 (MAC)

積和演算器 (MAC) は符号付数字の 8 ビット乗算器および総和用 32 ビット加算器を提供しています。MAC は演算の実行やデジタル フィルターの実装に非常に便利です。デバイスの特定のレジスタに読み書きすることで MAC を使用できます。乗算の後、値をレジスタから読み出すかアキュムレータに保存できます。クリア レジスタ (MACx_CL1 と MACx_CL2) に書き込むことでアキュムレータをクリアし、0 の値にリセットすることができます。詳細については、PSoC 1 テクニカル リファレンス マニュアルの第 26 章を参照してください。

4.4.4 電圧リファレンス

PSoC 1 には電圧リファレンス向けに多くのオプションが用意されています。次は 3 つの主な用語です。

1. AGND
2. RefHi
3. RefLo

デバイスのアナログ信号はアナログ グランド (AGND) にバイアスされています。AGND の電圧位置は開発者によって幅広いオプションから選択されます。AGND より高いアナログ信号は正電圧、それ未満は負電圧とされます。RefHi と RefLo はアナログ システムの上限と下限を意味します。詳細については、AN2219 - PSoC® 1 Selecting Analog Ground and Reference を参照してください。

4.4.5 専用 I^2C ハードウェア

I^2C 通信ブロックは、PSoC デバイスを 2 線式 I^2C シリアル通信バスとインターフェースするよう設計されたシリアル-パラレル プロセッサです。M8C マイクロコントローラーの過剰な介入とオーバーヘッドをなくすために、このブロックは I^2C 固有のステータス検出とフレーミング ビット生成のサポートを提供します。

以下は PSoC I^2C ハードウェア コントローラーの主な特長と機能です。

- 業界標準の Philips® I^2C バス互換インターフェース
- マスターおよびスレーブ動作、マルチマスター可能
- 2 本のピン (SDA と SCL) のみで I^2C バスとのインターフェースを実現
- 100/400kbps の標準データ転送速度;50kbps にも対応
- 7 ビット アドレッシング モード;10 ビット アドレッシングにも対応

I^2C の使用の詳細情報は、「AN50987: Getting Started with I2C in PSoC 1」を参照してください。

4.5 GPIO システム

GPIO システムは CPU やペリフェラルと外部間のインターフェースを提供します。各 GPIO は、デジタル I/O として使用されると、ピン当たり 10mA の吐き出し電流、25mA の吸い込み電流が可能です。デバイスは片側 100mA、合計 200mA を吸い込めます。したがって、任意のポートの偶数番号ピンは合計 100mA、奇数番号ピンはもう 100mA の電流に対応できます。デバイスは片側 40mA、合計 80mA を吐き出せます。詳細については、PSoC 1 テクニカル リファレンス マニュアルの第 6 章および「AN2094 - PSoC® 1 - Getting Started with GPIO」を参照してください。

4.6 CapSense

PSoC 1 デバイスには、CapSense として知られた静電容量タッチ センシングに対応しているデバイスがあります。静電容量タッチ センサーは、人体の静電容量を使ってセンサー上またはセンサー近く (近接) の指の存在を検出するユーザー インターフェースのデバイスです。静電容量センサーは審美的に優れており、使い易く、長寿命を持っています。CapSense により、高価で信頼性の低いメカニカル ボタンを、単にプリント基板上の銅配線である静電容量ボタンに置き換えることができます。CapSense はボタンやスライダー、トラックパッド、近接センサーなど幅広いセンサーに対応しています。CapSense に対応している PSoC 1 の一覧は、表 1 を参照してください。詳細については、「AN64846 - Getting Started with CapSense」を参照してください。

4.7 ダイナミックなリコンフィギュアビリティ

PSoC 1 では、最終製品の詳細が分かっていなくてもプロジェクトの設計を開始して、必要に応じてリソースを追加することができます。デジタルまたはアナログ リソースが利用可能である限り、調整、適用、拡張が可能です。リソースが消費されてしまった設計でも、別の機能の実行するために実行中に PSoC をダイナミックにリコンフィギュレーションすることができます。これにより、PSoC のシステム リソースを再利用し、集積能力を最大限に発揮させることができます。詳細については、「AN2104 - PSoC®1 - Dynamic Reconfiguration With PSoC® Designer™」を参照してください。

5 開発ツール

サイプレスの PSoC 1 は、PSoC Designer ソフトウェア ツール、開発キットやプログラミング ハードウェア、デバッグ ハードウェアを含むハードウェア ツールなどの品質の高い開発ツールでサポートされています。これらのツールにより、デバイスのコンフィギュレーション、アプリケーション コードの開発、また組み込み設計のビルド、デバッグ、そして展開が可能です。

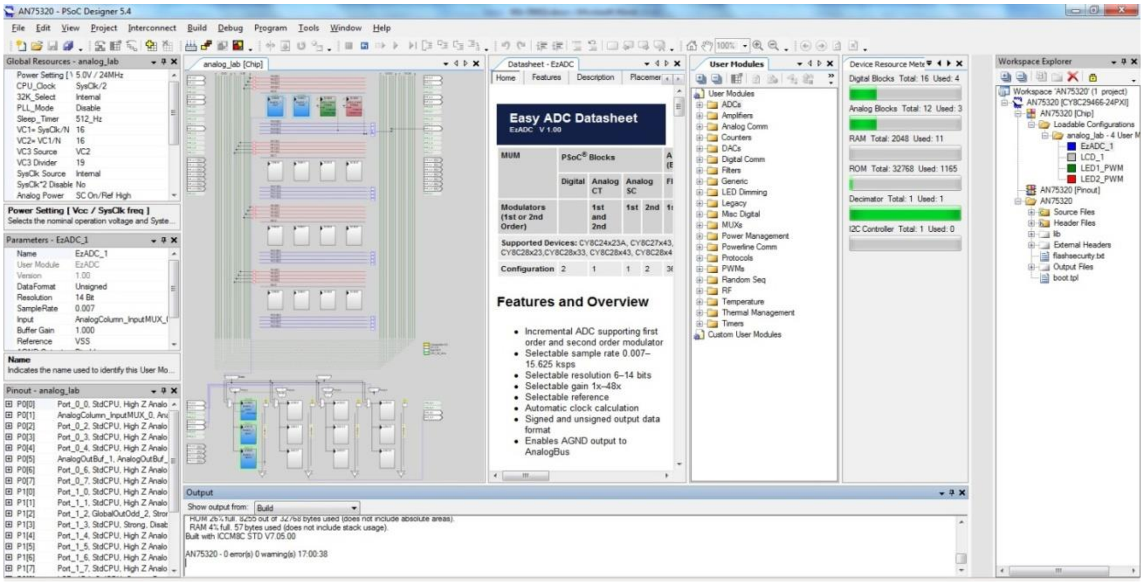
5.1 ソフトウェア: PSoC Designer IDE

サイプレスの PSoC Designer は、PSoC 1 デバイスのカスタマイズ、コンフィギュレーション、プログラミングに使用される統合開発環境 (IDE) です。このツールは http://www.cypress.com/go/psocdesigner からダウンロードできます。

PSoC Designer は、アナログとデジタル ペリフェラルをコンフィギュレーションして PSoC アプリケーションを作成し、アプリケーションコードを書き、本アプリケーション ノートで説明するその他の機能を実行することが可能な完全な開発環境を提供します。さらにこのソフトウェアにより、PSoC ICE-Cube デバッグ プラットフォームを使用して PSoC デバイスをプログラムし、プロジェクトをデバッグすることが可能です。

図 5 は、PSoC Designer のレイアウトおよび IDE の各部分の説明を示します。なお、これは PSoC Designer のデフォルトレイアウトではありません。Designer を起動する時に以下に示すウィンドウが表示されない場合は、ツールバーの **View** ドロップダウン メニューから追加できます。

図 5. PSoC Designer のレイアウト



5.2 ハードウェア

サイプレスは設計の要件を満たすために PSoC 1 向けのハードウェア キットを提供しています。PSoC 1 デバイスを選択するには、「PSoC 1 Kits」および「PSoC 1 Kit Selector Guide」を参照してください。以下の節は開発キット、プログラミング ハードウェアおよびデバッグ ハードウェアの一覧について紹介します。

5.2.1 開発キット

サイプレスは PSoC 1 設計を容易に試作する手助のためにいくつの開発キットを提供しています。表 2 は PSoC 1 の評価をサポートするいくつかの開発キットを一覧にします。開発キット／基板の完全なリストについては、開発キットを参照してください。

表 2. PSoC 1 開発キット

開発キット	説明
CY3210-PSOCEVAL1	CY3210 PSoC Eval1 キットにより、サイプレスの PSoC 1 プログラマブル システムオンチップの設計手法とアーキテクチャの評価および実験が可能です。
CY8CKIT-001	CY8CKIT-001 PSoC 開発キットは、異なるソリューションの試作と評価を可能にする一般的な開発プラットフォームを提供します。

5.2.2 プログラミング ハードウェア

サイプレス提供およびサードパーティのツールのいくつかのオプションから選択できます。サイプレスは 4 種のプログラミング デバイスを提供しています: MiniProg1、MiniProg3、ICE-Cube、CY3207-ISSP。

MiniProg1

MiniProg1 は多くの PSoC 1 キットに同梱されている ISSP プログラマです。このプログラマは CY25/26xxx を除き、すべての PSoC 1 デバイスに使用できます。MiniProg1 は PSoC 3 および PSoC 5 デバイスのプログラミングには使用できません。

MiniProg3

MiniProg3 は CY8CKIT-001 開発キットに同梱されている ISSP プログラマです。MiniProg3 には、PSoC 1、PSoC 3 および PSoC 5 デバイス用のプログラマ、PSoC 3 および PSoC 5 アーキテクチャ用のデバッグ ツール、I2C シリアル接続のデバッグおよび PSoC デバイスとの通信用の USB-I2C ブリッジが一体化されています。

サードパーティのプログラマ

サードパーティのプログラマ ツールの一覧は、http://www.cypress.com/documentation/development-kitsboards/general-psoc-programming を参照してください。これらのツールは PSoC 1 デバイスのプログラミングをサポートするように設計およびテストされ、サイプレスによって認定されました。

5.2.3 デバッグ ハードウェア

CY3215A-DK

CY3215A-DK In-Circuit Emulation Lite Development Kit にはインサーキット エミュレータ (ICE) が含まれています。ICE-Cube は主に PSoC 1 デバイスのデバッグに使用しますが、ISSP を使用して PSoC 1 デバイスのプログラムもできます。青色の CAT5 ケーブルやフレックス ケーブルを使ってデバッグ ポッドとインターフェースする代わりに、黄色の ISSP ケーブルを使ってデバイスをプログラムすることができます。このため、ICE-Cube はデバッグとプロダクション環境に非常に便利です。詳細は、「AN73212 – Debugging with PSoC 1」を参照してください。

6 PSoC 1 学習資料

本節は、PSoC 1 の入門および完全なアプリケーションの開発に役立つ PSoC 1 学習資料の一覧を提供します。データシート、リファレンス マニュアル、ビデオ、アプリケーション ノートなど PSoC 1 開発者が利用できる多くのリソースが用意されています。

6.1 PSoC 1 データシート

全ての PSoC 1 デバイス ファミリの機能および電気的仕様をリストアップする PSoC 1 データシートについては、PSoC 1 データシートのページにアクセスしてください。

6.2 PSoC 1 Designer の学習

PSoC Designer ホームページへアクセスして PSoC Designer の最新版をダウンロードしてください。

PSoC Designer を起動して次の項目に移動します。

PSoC Designer ユーザー ガイド: PSoC Designer ユーザー ガイドのページへアクセスしてください。

6.2.1 IDE ユーザー ガイド:

PSoC Designer を起動して、**Help** > **Documentation** > **Designer Specific Documents** > **IDE User Guide** を選択します。このガイドは PSoC Designer プロジェクトを開発するための基礎知識を提供します。

6.2.2 サンプル プロジェクト

PSoC Designer を起動して、**Design Catalog** > **Example projects** を選択します。これらのサンプル プロジェクトは、PSoC Designer のコンポーネントのコンフィギュレーションと使用方法をデモします。

6.2.3 PSoC Designer のトレーニング

これらのトレーニングは設計の手助けとなり、「Intro to PSoC Designer」、「Dynamic Reconfiguration」および「Debugging with PSoC」の内容を演習します。

6.2.4 コンポーネント データシート:

PSoC Designer を起動して、**Workspace Explorer** を開きます。Chip タブで、コンポーネントを右クリックして Datasheet を選択します。

6.3 アプリケーション ノート

アプリケーション ノートは、デバイスの特定の機能を理解し、PSoC 1 アプリケーションを設計する手助けをします。PSoC 1 のアプリケーション ノートの完全なリストについては、PSoC 1 アプリケーション ノートのページへアクセスしてください。

6.4 知識ベース記事

技術サポート関連のよく聞かれる質問と答えを蓄積したデータベースがあります。PSoC 1 の知識ベース記事の完全なリストについては、知識ベース記事のページへアクセスしてください。

6.5 テクニカル リファレンス マニュアル (TRM)

TRM は PSoC 1 デバイスの内部アーキテクチャの詳細な説明を提供します。PSoC 1 の TRM の完全なリストについては、PSoC 1 テクニカル リファレンス マニュアルのページへアクセスしてください。

6.6 デバイス エラッタ

これらの資料は、デバイス データシートまたはテクニカル リファレンス マニュアルから外れる仕様 (ある場合) を記載しています。デバイスのエラッタ資料の一覧については、デバイス エラッタのページへアクセスしてください。

6.7 テクニカル サポート

お問い合わせまたはご質問は、弊社のテクニカル サポート チームが対応いたします。サイプレスのテクニカル サポートにアクセスし、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。

米国のお客様は、弊社フリーダイヤル (+1-800-541-4736 ext.8) までお電話いただければ、弊社のテクニカル サポート チームがご対応いたします。

早急な対応が求められる場合には、下記の方法をご利用ください。

- セルフ ヘルプ
- 所在地の販売代理店

7 はじめての PSoC 1 設計

本節は、PSoC Designer を用いて PSoC 1 を使ったシンプルなデザインを設計して、それを PSoC デバイスにプログラムし、また PSoC 1 デモ基板をコンフィギュレーションしてアプリケーションの結果を検証する、段階を追ったプロセスを説明します。

7.1 設計について

本設計では、2 つの GPIO で 2 つの PWM 出力 (1 つはハードウェア リソースを使用し、もう 1 つはソフトウェアを使用) を生成するシンプルなプロジェクトを作成します。1 つの GPIO は固定速度で点滅する LED を示し、もう 1 つの GPIO はパルス (ハートビート) LED を示します。本設計ではまた、ADC ユーザー モジュールを使用してアナログ電圧を測定します。

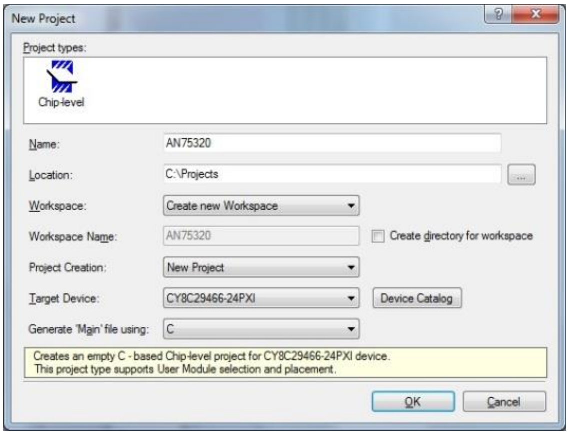
7.2 はじめての PSoC 1 設計の作成

7.2.1 プロジェクトの作成

まず、PSoC Designer でプロジェクトを作成します。ここで、デバイスを選択して、開発言語を C に設定します。この例では、CY3210-PSoCEval1 キットに同梱されている CY8C29466-24PXI を使用します。

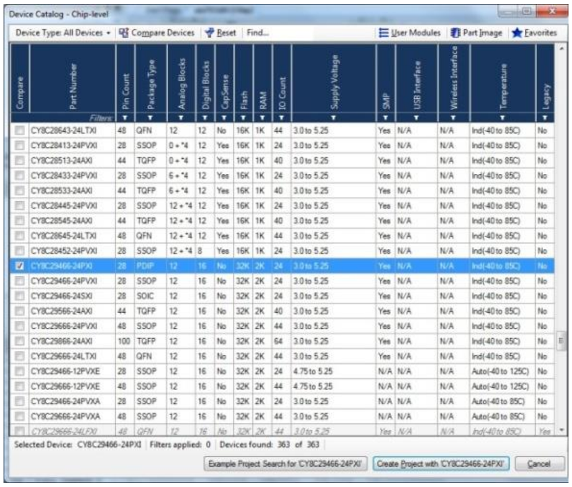
1. 図 6 に示すように、PSoC Designer で **File** > **New Project** を選択して *AN75320* と名付けます。

図 6. PSoC Designer で新規プロジェクトを作成



2. 図 7 に示すように、**Device Catalog** をクリックしてデバイスを選択します。

図 7.「Device Catalog」



注: このプロジェクトを他のキットで開発する際は、以下の製品番号を使用してください。

- CY3214-PSoCEvalUSB: CY8C24994-24LTXI
- CY3271-PSoC First Touch: CY8C20634-12FDXI

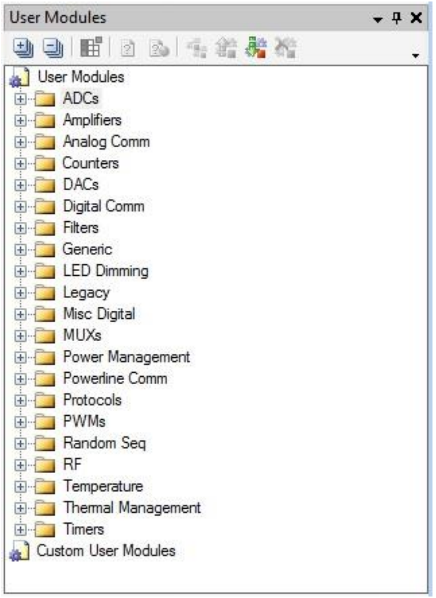
3. 「**Create Project with CY8C29466-24PXI**」をクリックして、**OK** をクリックします。

プロジェクト ワークスペースが開き、Chip Editor と Workspace Explorer を表示します。Chip Editor は、選択したデバイスのブロック図を示します。Ctrl を押しながらクリックして Chip Editor ビューをズームインします。

7.2.2 ユーザー モジュールの選択

このステップでは、User Module Catalog から必要なコンポーネントを選択してそれらをコンフィギュレーションします。ユーザモジュールのウィンドウは図 8 に示すように、右下にあります。**View** > **User Module Catalog** を選択してそれを表示させます。

図 8. ユーザ モジュール カタログ ウィンドウ



このプロジェクトは次のユーザ モジュールを使用します。

- ADC ユーザー モジュール 1 個
- PWM8 ユーザー モジュール 2 個
- LCD ユーザー モジュール 1 個

1. ユーザ モジュールを配置するするには次のステップを行ってください：User Module Catalog 内で **ADCs** フォルダを展開します。
2. **EzADC** を右クリックして、Place を選択します。EzADC コンフィギュレーション ウィンドウが開きます。
3. コンフィギュレーション ウィンドウ内で Double Stage Modulator を選択して、OK をクリックします。(このコンフィギュレーションでは、高い性能のために 2 個のスイッチト キャパシタを使用します。)
4. PWMs フォルダの下で PWM8 を右クリックして、Place を選択します。
5. 配置プロセスを繰り返して 2 個の **PWM8** ユーザー モジュールを設計に配置します。
6. Misc Digital フォルダから **LCD** をクリックして、**Place** を選択します。

必要なユーザー モジュールの配置を完了した後、Designer ウィンドウは図 9 のようになります。また、配置したユーザーモジュールは Workspace Explorer に表示されます。配置した各ユーザー モジュールは ComponentName_1 などのように名付けられます。例えば、EzADC ユーザー モジュールは EzADC_1 としてリストアップされます。

図 9. Chip Editor 内の EzADC と PWM ユーザー モジュールの配置 (デジタル ブロック)

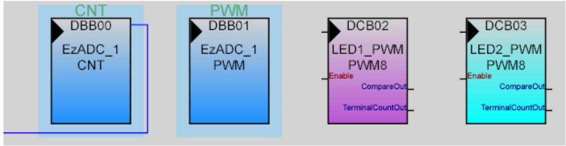
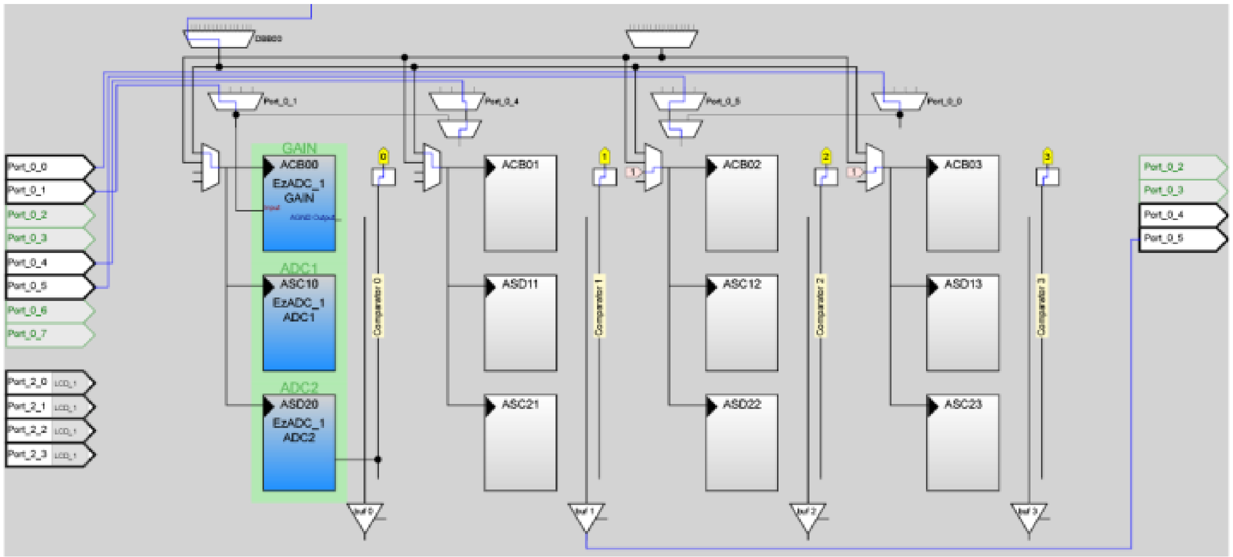


図 10. Chip Editor 内の EzADC ユーザー モジュールの配置 (アナログ ブロック)



7.2.3 ユーザー モジュール パラメーターの設定

このステップでは、設計に配置したユーザー モジュールのパラメーターを設定します。

Workspace Explorer または Chip Editor からユーザー モジュールをクリックすると、選択したユーザー モジュールの Parameters ウィンドウが左側に表示されます。このウィンドウでは、パラメーターを変更することができます。

設定または変更するパラメーターについては、以下の図を参照してください。

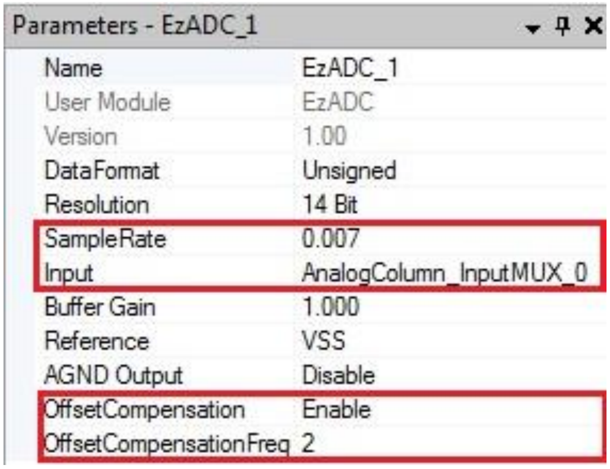
1. **EzADC_1** を選択して、そのパラメーターを表 3 に基づいて変更します。変更した後、ウィンドウは図 11 のようになります。初期設定値から変更されるパラメーターは強調表示されています。

表 3. EzADC_1 ユーザー モジュールのパラメーター

パラメーター	値	説明
Name (名称)	EzADC_1	ユーザー モジュールに名称を付ける
DataFormat (データ形式)	Unsigned (符号なし)	入力は正の値のみであるため、出力を符号なしデータに設定
Resolution (分解能)	14 Bit (14 ビット)	EzADC の最大分解能を設定
Sample Rate (サンプル レート)	0.007	サンプル レートは分解能に依存。EzADC のユーザー モジュール データシートを参照してください。
Input (入力)	AnalogColumn_InputMUX_0	ソースの入力アナログ電圧を受信するようアナログ入力データ バスを選択
Offset Compensation (オフセット補正)	Enable (有効)	相関 2 重サンプリングを有効にする

パラメーター	値	説明
Buffer Gain (バッファ ゲイン)	1.000	増幅しないように PGA ゲインを 1 に設定
Reference (リファレンス)	VSS	オフセット補正を行うために PGA のリファレンス電圧を設定
AGND Output (AGND 出力)	Disable (無効)	AGND 出力を無効にする。このオプションは、内部 AGND を外部ピンに出して両極性信号を ADC に接続するために使用可能
Offset Compensation Freq (オフセット補正周波数)	2	オフセットの補正間の ADC サンプル数を設定

図 11. EzADC_1 ユーザー モジュールのパラメーター

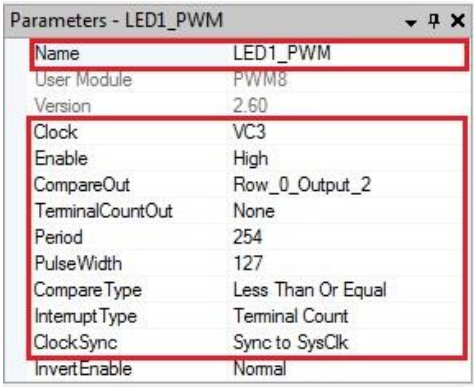


PWM8_1 を変更して、そのパラメーターを表 4 のように変更します。変更した後、ウィンドウは図 12 のようになります。これで、ユーザー モジュールの名称が **LED1_PWM** に変更されたことに注意してください。

表 4. LED1_PWM ユーザー モジュールのパラメーター

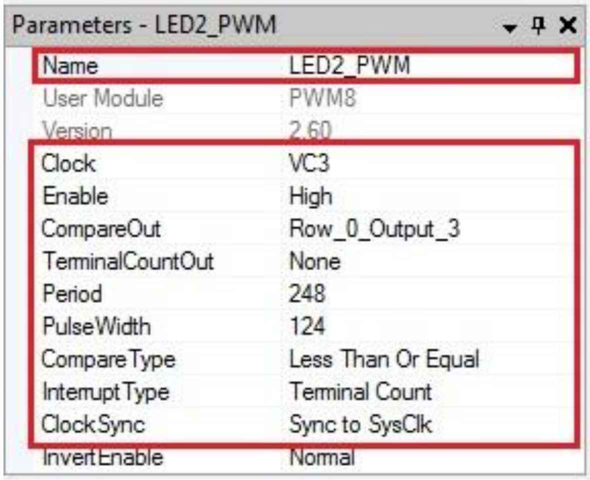
パラメーター	値	説明
Name (名称)	LED1_PWM	ユーザー モジュールに名称を付ける
Clock (クロック)	VC3	クロック ソースに **VC3** を選択
Enable (イネーブル)	High (高)	連続カウントのために「High」を選択
CompareOut (比較出力)	Row_0_Output_2	行出力バスを介して比較出力を GPIO に接続
Terminal Count Out (ターミナル カウント出力)	None (無)	これは補助カウンターであり、どの行出力バスにも接続する必要はない
Period (周期)	254	カウンターの周期を設定
Pulse Width (パルス幅)	127	PWM 出力のパルス幅 (50%デューティ サイクル) を設定
Compare Type (比較条件)	Less Than Or Equal (未満または等しい)	比較機能の比較条件を選択
Interrupt Type (割り込みタイプ)	Terminal Count (ターミナル カウント)	これを選択すると、カウンター レジスタのターミナル カウント時に割り込みが発生される
Clock Sync (クロック同期)	Sync to SysClK (SysClK と同期)	クロック入力をシステム クロックと同期化
Invert Enable (イネーブル入力の反転)	Normal (通常)	イネーブル入力信号の極性を決定

図 12. PWM_1 ユーザー モジュールのパラメーター



2. **PWM8_2** を選択して、そのパラメーターを図 13 のように変更します。これで、ユーザー モジュールの名称が **LED2_PWM** に変更されたことに注意してください。

図 13. PWM_2 ユーザー モジュールのパラメーター

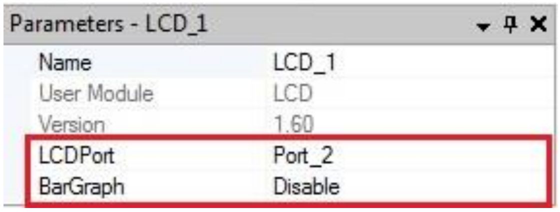


3. **LCD**_1 ユーザー モジュールを変更して、そのパラメーターを表 5 のように変更します。変更した後、ウィンドウは図 14 のようになります。

表 5. LCD_1 ユーザー モジュールのパラメーター

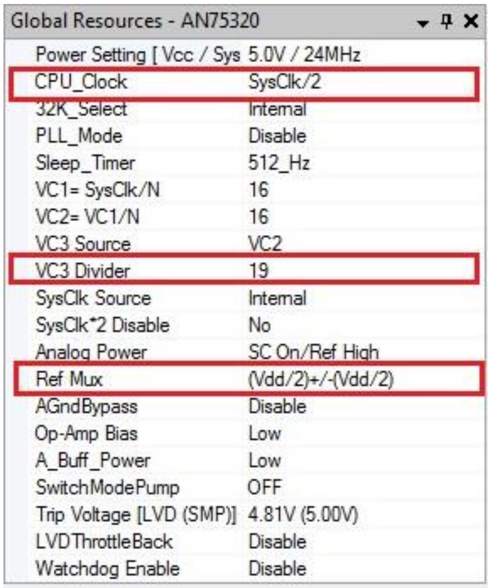
パラメーター	値	説明
Name (名称)	LCD_1	ユーザー モジュールに名称を付ける
LCD Port (LCD ポート)	Port_2	LCD ディスプレイ モジュールを割り当てられたポートへインターフェースさせる
BarGraph (バー グラフ)	Disable (無効)	バー グラフ機能を無効にする

図 14. LCD パラメーター



4. **View > Global Resources** を選択してグローバル リソース ウィンドウを開きます。ここで、ソース クロック VC3 を PWM クロック周波数に、Ref Mux を ADC 入力データ範囲に設定します。他のパラメーターは初期値のままにします。変更した後、グローバル リソース ウィンドウは図 15 のようになります。グローバル リソースの詳細情報については、IDE ユーザー ガイドを参照してください。

図 15. グローバル リソースのパラメーター



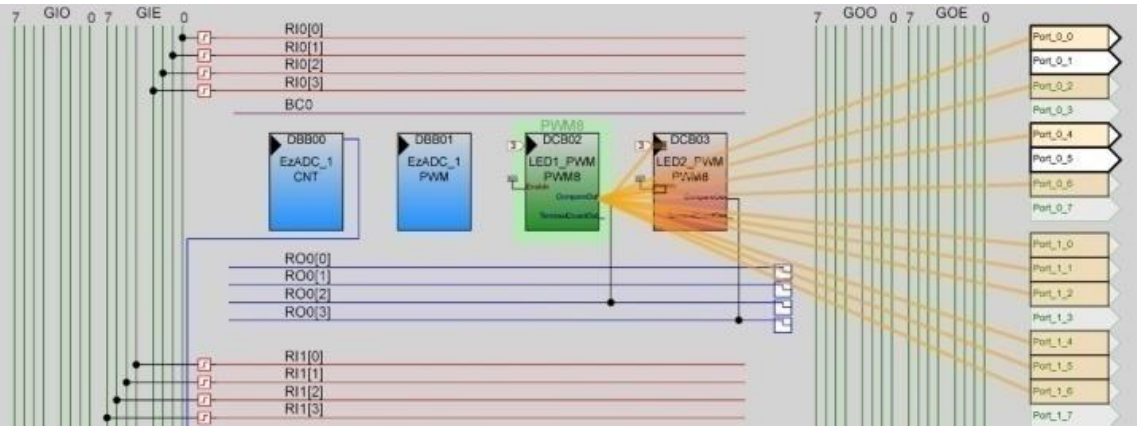
7.3 PWM 信号のルーティング

本節では、PWM 信号を PWM ユーザー モジュールから選択された GPIO へ配線します。 GPIO ピン上の信号を見るには、PWM 信号の配線を実装しなければなりません。

本設計はまた、2 つの PWM 出力を XOR ゲートに供給して論理レベルを行出力に実装します。XOR がとられる 2 つの PWM 信号の周期差は、パルス (ハートビート) エフェクトを作り出します。

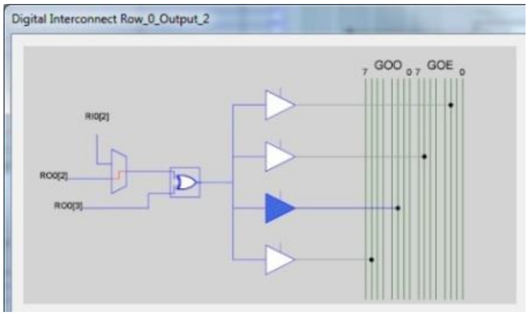
1. LED1_PWM の CompareOut を Row_0_Output_2 に接続します。
2. LED2_PWM の CompareOut を Row_0_Output_3 に接続します。
3. 自動ルーティングを使用して LED1_PWM の CompareOut を Port_1_2 に接続します。**Shift** を押しながら LED1_PWM の CompareOut ピンを選択して、図 16 に示すようにそれを Port_1_2 と接続します。また、GlobleOutOdd_2 を使用して LED1_PWM を Port_1_2 に手動で接続できます。

図 16. 行出力に接続された PWM



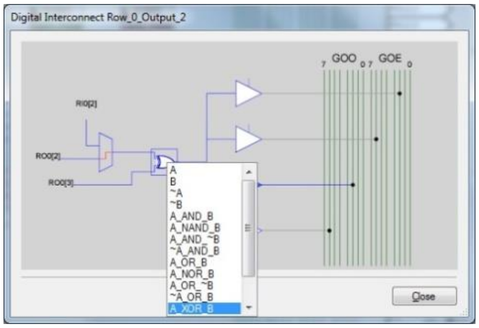
4. Row_0_Output_2 デジタル相互接続をクリックして、図 17 に示すようにコンフィギュレーション オプションを開きます。

図 17. RO0[2]のデジタル相互接続ビュー



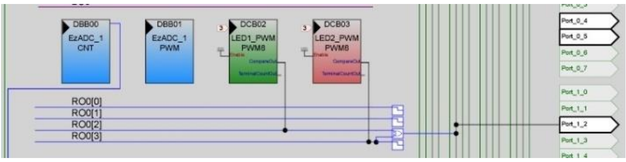
5. LogicTable_Select（中央の四角形ボックスに示される）をクリックし、図 18 に示すようにドロップダウン リストから A_XOR_B を選択して、**Close** をクリックします。

図 18. デジタル相互接続のコンフィギュレーション



6. デジタル ルーティング図が図 19 のようになったことを確認してください。それらが一致していれば、パルス状のハートビート LED のコンフィギュレーションは完了しました。

図 19. Designer のチップ ビュー



7.4 コードの追加

このステップでは、点滅 LED を生み出すソフトウェア LED を実装するためにコードを追加して全てのユーザー モジュールと GPIO をコンフィギュレーションします。以下のコードをプロジェクトの *main.c* ファイルに追加します。

1. Workspace Explorer で、**AN75320 Folder** の下で **Source Files** フォルダを探して *main.c* ファイルを開きます。
2. 以下の C コードをコピーして *main.c* ファイルにある既存のコードを置き換えてペーストします。

コード 1. プロジェクト 1 の main.c

```
/* part specific constants and macros */
#include <m8c.h>

/* PSoC API definitions for all User Modules */
#include "PSoCAPI.h"

/* Definitions for all input and output operation */
#include "stdio.h"
```

```
/* Macros to set ADC parameters */
#define GAIN                1
#define MAX_ADC_COUNTS      16383
#define ADC_RANGE           5000

/* Macros to select port 1 */
#define PORT_1_3        PRT1DR

/* Variable to store the ADC result */
WORD adc_result;

/* Variable to store the measured input in millivolts */
WORD v_in;
void main(void)
{
    static unsigned int index;

   /* Buffer used for the long to ASCII conversion */
   char LCDBuffer[17];

   /* Initializes LCD to use the multi-line 4-bit interface  */
   LCD_1_Start();

   /* Enable global interrupts */
   M8C_EnableGInt ;

   /* Set the position to print the character */
   LCD_1_Position(0,0);

   /* print the Hello World in the first line */
   LCD_1_PrCString("Hello World!");

   /* Starts the LED1_PWM and LED2_PWM, high enable input begins the Counter */
   LED1_PWM_Start();
   LED2_PWM_Start();

   /* Initializes and starts the EzADC User Module resources */
   EzADC_1_Start(EzADC_1_HIGHPOWER);

   /* Starts the ADC conversion */
   EzADC_1_GetSamples(0);
             while(1)
{
                    /* Wait for the ADC result to be available */
                    while(!(EzADC_1_fIsDataAvailable()));

                    /* Read the ADC result and clear the data ready flag */
                    adc_result = EzADC_1_iGetDataClearFlag();

                    /* Calculate input voltage in mV */
                    v_in = (DWORD)adc_result*ADC_RANGE / MAX_ADC_COUNTS / GAIN;

                    /* Convert the input voltage to an ascii string */
                    csprintf(LCDBuffer, "ADC INPUT:%4dmV", v_in);

                    /* Set the position to print the character */
                    LCD_1_Position(1,0);

                    /* print the voltage in the second line */
                    LCD_1_PrString(LCDBuffer);
```

```
                /* Toggle the pin 1[3]*/
                PORT_1_3 ^= 0x08;

                /* Give some delay to view toggling effect*/
          for(index = 0; index < 22000; index++);
          }
}
```

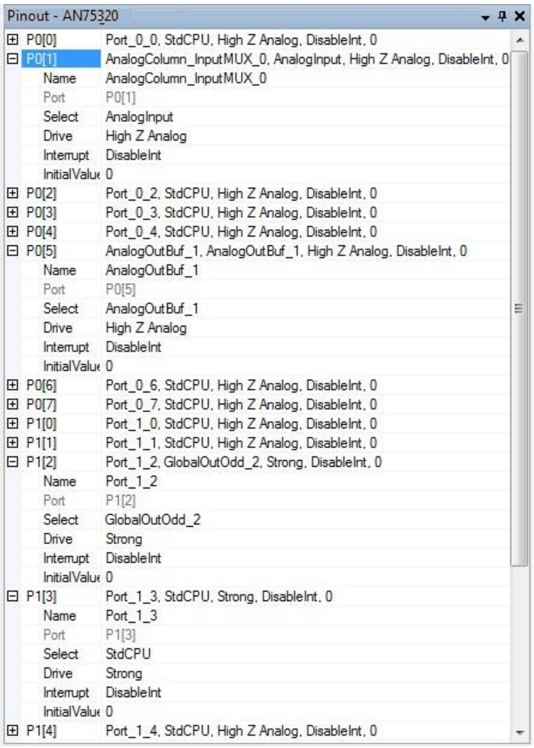
3. *local.mk* ファイルをプロジェクトに追加して ADC の浮動小数点を有効にします。この機能の詳細情報については、*local.mk* ファイルを見てください。

7.5 ピン配置のコンフィギュレーション

このステップでは、ピンが LED を駆動するよう GPIO ドライブ モードをコンフィギュレーションします。Pinout ビュー (**View** > **Pinout**) で、選択したデバイスのピン配置をコンフィギュレーションします。Pinout パネルはワークスペースの左下に表示されます。

1. Pinout ビューで、P0[1]、P0[5]、P1[2]、および P1[3]のオプションを展開します。図 20 に示すようにピンをコンフィギュレーションします。

図 20. ピン コンフィギュレーション

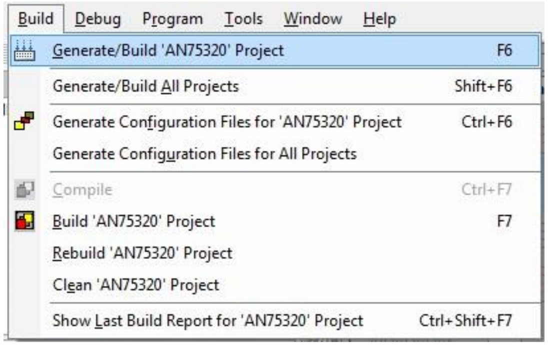


7.6 ビルドおよびプログラム

このステップでは、MiniProg1 デバイスをコンピュータに接続し、ビルドしたプロジェクトを PSoC デバイスにプログラムします。

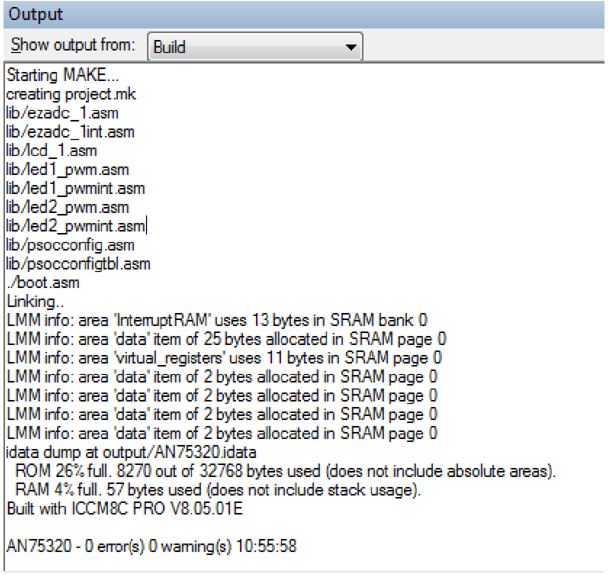
1. **Build** > **Generate/Build 'AN75320' Project** を選択する、または **F6** を押します (図 21 を参照してください)。

図 21. ビルドおよび生成オプション



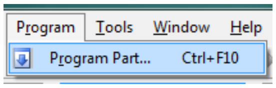
2. 図 22 に示すように、出力ウィンドウは RAM/フラッシュの使用量およびエラーと警告の件数を含むビルドの状態を示します。

図 22. 出力ウィンドウ



3. MiniProg1／MiniProg3 をコンピュータに接続します。
4. MiniProg1／MiniProg3 を基板のプログラミング ヘッダーに接続します。詳細については、**Error! Reference source not found.**のステップ 4 を参照してください。
5. **Program** > **Program Part** を選択します (図 23 を参照してください)。図 24 に示すように、Program Part ウィンドウが表示されます。

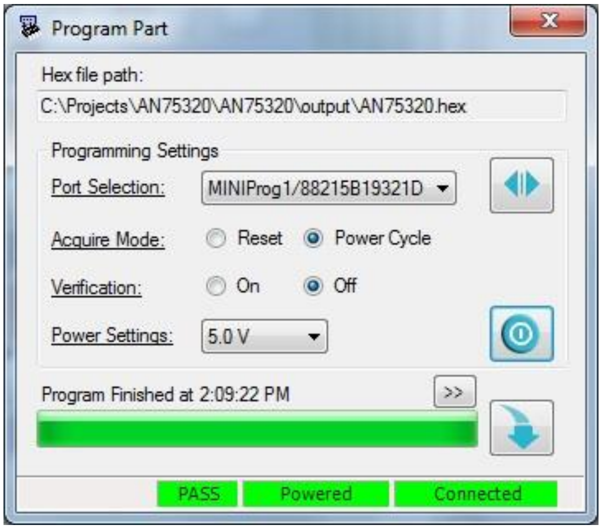
図 23. Program Part オプション



6. Program Part ウィンドウで、以下を行ってください。
   a) **Connect** ボタン (Port Selection フィールドの隣にある) をクリックしてデバイスに接続します。
   b) MiniProg がデバイスへ電源を供給していて、デバイスへの電源を一旦切ってすぐに入れなおすことでデバイスを選択する場合、**Acquire Mode** を *Power Cycle* に設定します。これは初期設定のオプションです。このプロジェクトでは、初期設定のオプションを使用します。
   c) デバイスが外部から電源供給されていて、MiniProg がデバイスをリセットすることでのみデバイスを選択できる場合は、**Acquire Mode** を *Reset* に設定します。

7. **Program** ボタンを押して、図 24 のようにデバイスをプログラムします。

図 24. プログラミングの状態

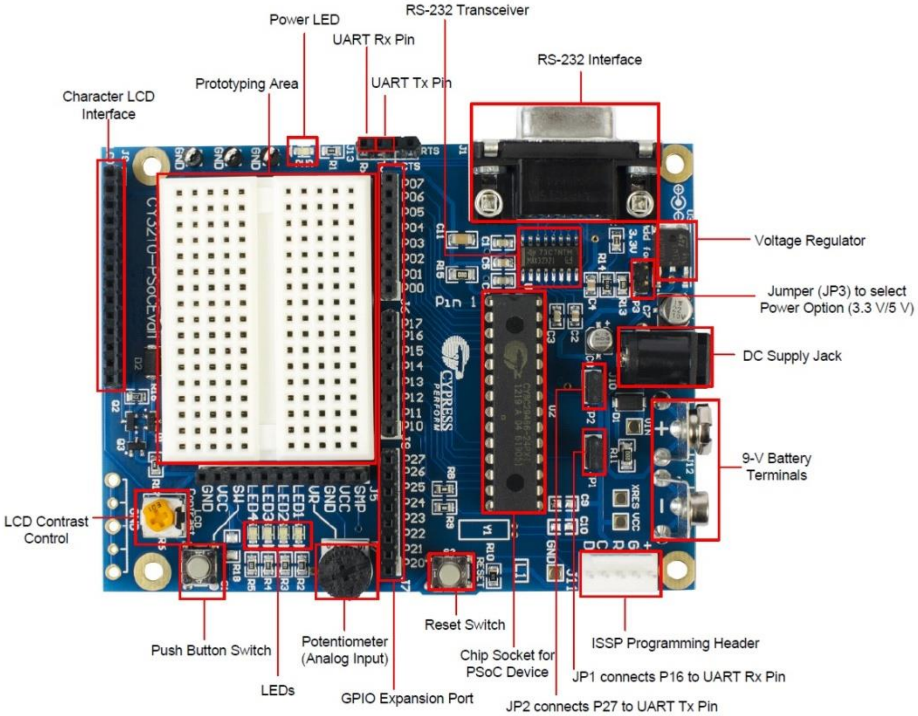


8. プログラミングが完了した後、**Power** ボタンを押してデバイスの電源を切り替えます。これで、プログラマは基板の電源を供給します。LED 点滅と退色を観察します。

7.7 CY3210-PSoCEval1 基板の設定

このデモは CY3210-PSoCEval1 ハードウェアと互換性があります。このキットに関する詳細は、http://www.cypress.com/documentation/development-kitsboards/cy3210-psoceval1 をご覧ください。

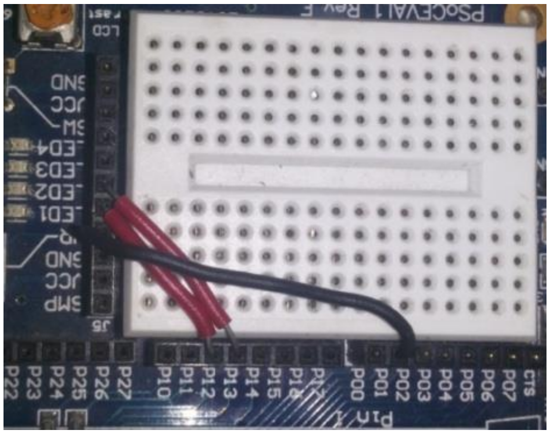
図 25. CY3210-PSoCEval1 キット



CY3210-PSoCEval1 キットをコンフィギュレーションしてプログラムする手順は以下の通りです。このキットには MiniProg や ICE-Cube などのプログラミング デバイスが必要であることに注意してください。

1. デモ基板にケーブルが一切接続されていない状態で、J1、J2 および J3 にはケーブルが差し込まれていないことを確認します。
2. P0[1]と VR 間を 1 本のワイヤで、P1[2]と LED1 間を 1 本のワイヤで接続し、P1[3]と LED2 間をもう 1 本のワイヤで接続します (図 26 を参照してください)。

図 26. CY3210-PSoCEval1 のピン接続



3. 基板に搭載されているデバイスが CY8C29466-24PXI であることを確認します。
4. 基板に電源を供給するためには、MiniProg 1／MiniProg 3 をプログラミング ヘッダー (J11) に接続します。同様に、12V の DC 電源をキット上の J10 に接続します。または、9V バッテリを J12 に接続することもできます。また、プログラミング デバイスを J11 に接続します。外部の電源や電池を使用する場合ではリセット モードを使用します。基板の接続は図 27 に示します。

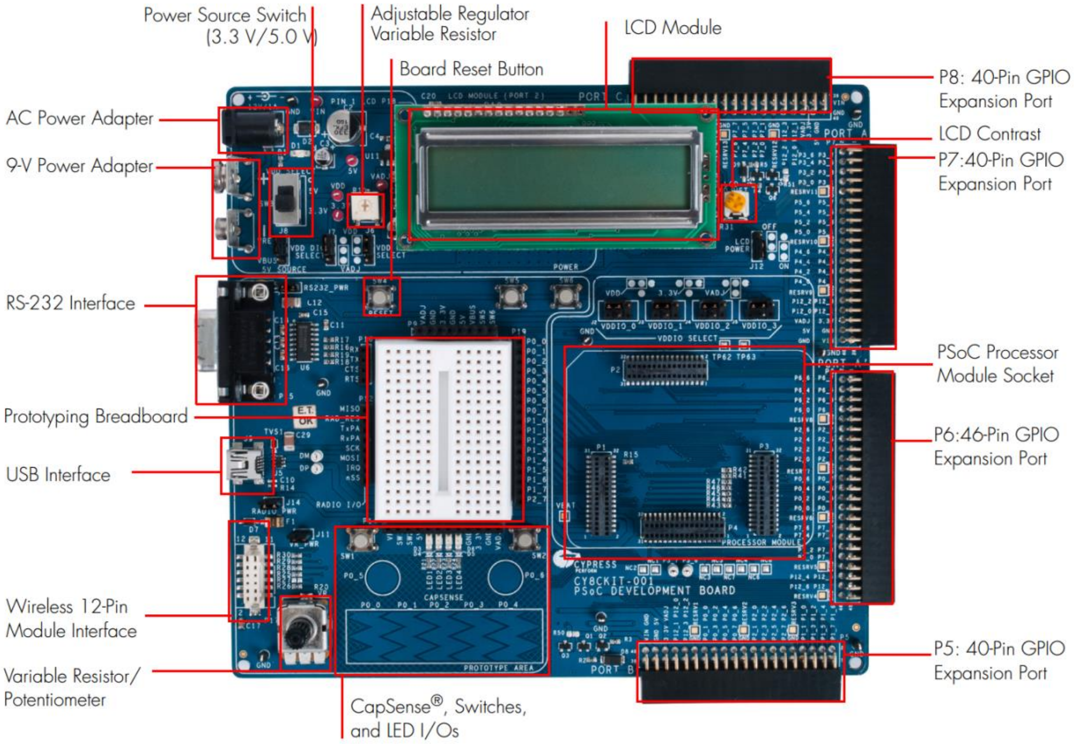
図 27. 電源とプログラム接続

7.8 CY8CKIT-001 ボードの設定

CY8C29x66 プロセッサ モジュール 1 個を使用すると、このデモは CY8CKIT-001 ハードウェアとも互換性があります。このキットに関する詳細は、http://www.cypress.com/?rID=37464 をご覧ください。

CY8CKIT-001 キットをコンフィギュレーションしてプログラムする手順は以下の通りです。このキットには MiniProg や ICE-Cube などのプログラミング デバイスが必要であることに注意してください。さらに、このキットは ICE-Cube を使用する際にオンチップ デバッグをサポートします。

図 28. CY8CKIT-001 キット



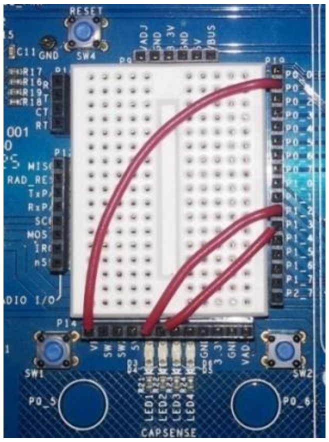
1. デモ基板に一切ケーブルが接続されていない状態で、表 6 のようにジャンパを設定して開発基板をコンフィギュレーションします。

表 6. ジャンパの設定

ジャンパ	設定
VDD 選択 (SW3)	5V (上の位置)
5V ソース (J8)	VREG (上ピン 2 本)
VDD デジタル (J7)	VDD (上ピン 2 本)
VDD アナログ (J6)	VDD (上ピン 2 本)
LCD 電源 (J12)	ON (下ピン 2 本)
VDDIO 選択 (J2～J5)	VDD (左上ピン 2 本)

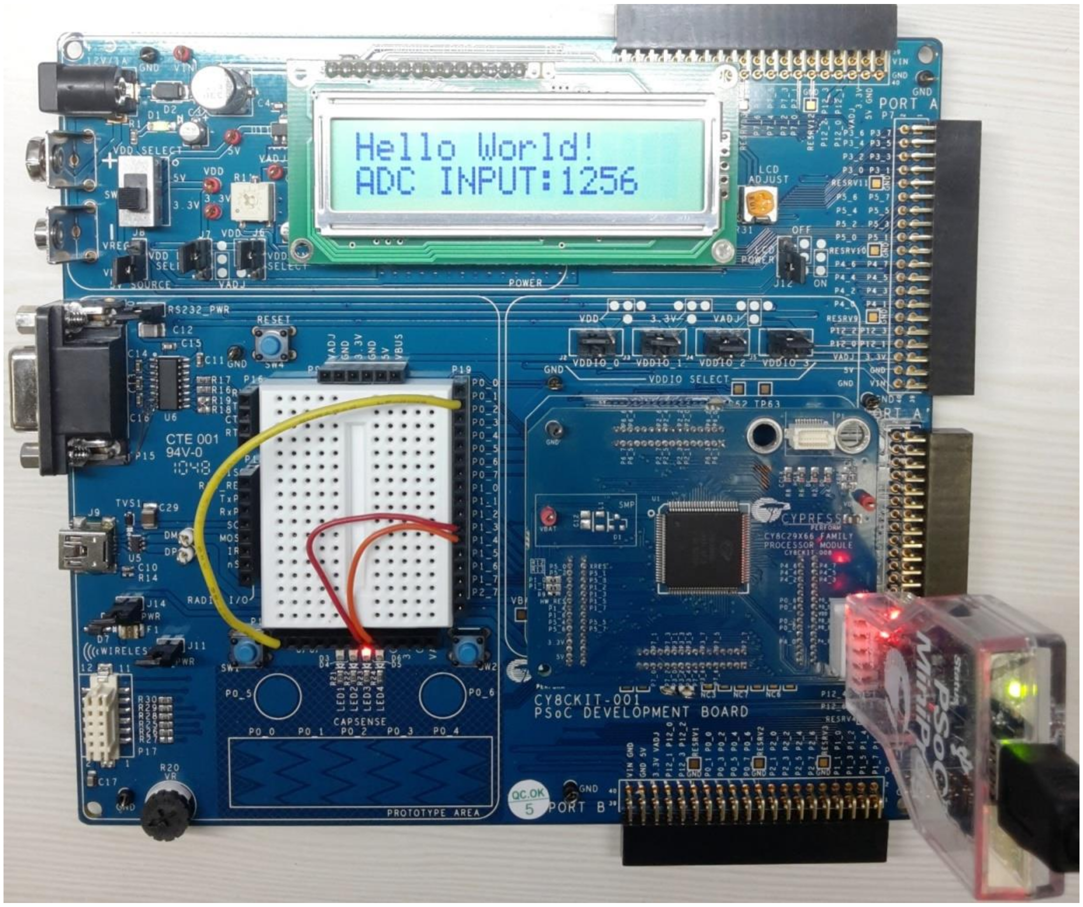
2. P0[1]と VR 間を 1 本のワイヤで、P1[2]と LED1 間を 1 本のワイヤで接続し、P1[3]と LED2 間をもう 1 本のワイヤで接続します (図 29 を参照してください)。

図 29. 別のピン接続図



3. MiniProg や ICE などのプログラミング デバイスを PSoC 1 プロセッサ モジュールの J5 に接続します。
4. 12V DC 電源か 9V バッテリから基板へ電源を供給するか、またはプログラマからの電源を使用します。外部の電源や電池を使用すると、リセット モードが使用されたことを確認してください。基板の接続については、図 30 を参照してください。

図 30. 電源とプログラム接続



アプリケーションのテストにいずれの開発キットを使用した場合でも、以下を確認することが可能です。

1. LCD ディスプレイが「Hello World」と「ADC INPUT:----mV」を表示します。
2. 1 個の LED が速く点滅します。
3. 2 個目の LED は、徐々に高くなり低くなる明度でパルスを発します。

8 まとめ

本アプリケーション ノートは PSoC 1 アーキテクチャおよび開発ツールを紹介しました。本書はまた、PSoC 1 プロジェクトを開始するのに必要なツールについて説明しました。PSoC 1 は、単一のチップにコンフィギュレーション可能なアナログとデジタル ペリフェラル機能、メモリ、M8C マイクロコントローラーを集積している真のプログラマブル組み込みシステムオンチップです。

本アプリケーション ノートはまた、PSoC 1 を迅速に深く理解できるように総合的なリソース収集へのアクセスを提供します。

説明した項目の詳細情報については、記載されたリンクや参照文献を参照する、または http://www.cypress.com/products/psoc-1 を訪問してください。

9 関連アプリケーション ノート

本アプリケーション ノートは PSoC 1 の基本的な情報を含んでいます。特定のトピックに関する詳細情報については、特定のアプリケーション ノートを参照してください。

- 入門
  AN54181 - Getting Started with PSoC 3
  AN79953 – Getting Started with PSoC 4
  AN77759 - Getting Started with PSoC 5LP
- アナログ
  AN2219 - PSoC® 1 Selecting Analog Ground and Reference
  AN74170 – PSoC 1 Analog Structure and Configuration with PSoC Designer™
  AN13666 - PSoC® 1 Driving Analog Buffer Output to the Rail
  AN2096 - PSoC® 1 - Using the ADCINC Analog to Digital Converter
- スイッチト キャパシタ ブロック
  AN2041 – Understanding PSoC 1 Switch Capacitor Analog Blocks
  AN2168 – PSoC 1 Understanding Switched Capacitor Filters
  AN16833 - Signal Mixing with PSoC® Switched Capacitor Blocks
- GPIO
  AN2094 - PSoC® 1 - Getting Started with GPIO
- プログラミング
  AN44168 - PSoC® 1 Device Programming using External Microcontroller (HSSP)
- デジタル
  AN2141 - PSoC® 1 Glitch Free PWM
- フラッシュ
  AN2015 - PSoC 1 Reading and Writing Flash & E2PROM
- I2C
  AN50987 – Getting Started with I2C in PSoC 1
- SPI
  AN51234 - Getting Started with SPI in PSoC® 1

- スリープ モード

  AN47310 - PSoC® 1 Power Savings Using Sleep Mode

- LCD

  AN56384 - PSoC® 1 Segment LCD Direct Drive
  AN2152 - PSoC® 1 Graphics LCD and PSoC® Interface

改訂履歴

文書名: AN75320 - PSoC® 1 入門

文書番号: 001-79363

版	ECN 番号	変更者	発行日	変更内容
**	3620170	HZEN	05/17/2012	これは英語版 001-75320 Rev. **を翻訳した日本語版 001-79363 Rev. **です。
*A	3889056	HZEN	01/29/2013	これは英語版 001-75320 Rev. **を翻訳した日本語版 001-79363 Rev. *A です。 1 ページの AN 番号のエラーを訂正しました。
*B	4669800	HZEN	03/27/2015	これは英語版 001-75320 Rev. *D を翻訳した日本語版 001-79363 Rev. *B です。
*C	4773026	HZEN	05/27/2015	変化なし日没 ECN ありません
*D	5045452	HZEN	12/24/2015	これは英語版 001-75320 Rev. *E を翻訳した日本語版 001-79363 Rev. *D 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

車載	cypress.com/go/automotive
クロック＆バッファ	cypress.com/go/clocks
インターフェース	cypress.com/go/interface
照明＆電源管理	cypress.com/go/powerpsoc
メモリ	cypress.com/go/memory
PSoC	cypress.com/go/psoc
タッチ センシング	cypress.com/go/touch
USB コントローラー	cypress.com/go/usb
ワイヤレス／RF	cypress.com/go/wireless

PSoC ソリューション

psoc.cypress.com/solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

サイプレス開発者コミュニティ

コミュニティ | フォーラム| ブログ| ビデオ| トレーニング

テクニカル サポート

cypress.com/go/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、各社の所有物です。

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

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 2012-2015. 本文書に記載される情報は予告なく変更される場合があります。Cypress Semiconductor Corporation (サイプレス セミコンダクタ社) は、サイプレス製品に組み込まれた回路以外のいかなる回路を使用することに対して一切の責任を負いません。サイプレス セミコンダクタ社は、特許またはその他の権利に基づくライセンスを譲渡することも、または含意することもありません。サイプレス製品は、サイプレスとの書面による合意に基づくものでない限り、医療、生命維持、救命、重要な管理、または安全の用途のために使用することを保証するものではなく、また使用することを意図したものでもありません。さらにサイプレスは、誤動作や故障によって使用者に重大な傷害をもたらすことが合理的に予想される生命維持システムの重要なコンポーネントとしてサイプレス製品を使用することを許可していません。生命維持システムの用途にサイプレス製品を供することは、製造者がそのような使用におけるあらゆるリスクを負うことを意味し、その結果サイプレスはあらゆる責任を免除されることを意味します。

このソースコード (ソフトウェアおよび／またはファームウェア) はサイプレス セミコンダクタ社 (以下「サイプレス」) が所有し、全世界の特許権保護 (米国およびその他の国)、米国の著作権法ならびに国際協定の条項により保護され、かつそれらに従います。サイプレスが本書面によりライセンシーに付与するライセンスは、個人的、非独占的かつ譲渡不能のライセンスであり、適用される契約で指定されたサイプレスの集積回路と併用されるライセンシーの製品のみをサポートするカスタム ソフトウェアおよび／またはカスタム ファームウェアを作成する目的に限って、サイプレスのソース コードの派生著作物をコピー、使用、変更そして作成するためのライセンス、ならびにサイプレスのソース コードおよび派生著作物をコンパイルするためのライセンスです。上記で指定された場合を除き、サイプレスの書面による明示的な許可なくして本ソース コードを複製、変更、変換、コンパイル、または表示することはすべて禁止します。

免責条項: サイプレスは、明示的または黙示的を問わず、本資料に関するいかなる種類の保証も行いません。これには、商品性または特定目的への適合性の黙示的な保証が含まれますが、これに限定されません。サイプレスは、本文書に記載される資料に対して今後予告なく変更を加える権利を留保します。サイプレスは、本文書に記載されるいかなる製品または回路を適用または使用したことによって生ずるいかなる責任も負いません。サイプレスは、誤動作や故障によって使用者に重大な傷害をもたらすことが合理的に予想される生命維持システムの重要なコンポーネントとしてサイプレス製品を使用することを許可していません。生命維持システムの用途にサイプレス製品を供することは、製造者がそのような使用におけるあらゆるリスクを負うことを意味し、その結果サイプレスはあらゆる責任を免除されることを意味します。

ソフトウェアの使用は、適用されるサイプレス ソフトウェア ライセンス契約によって制限され、かつ制約される場合があります。