

PSOC™ 4 MCU 入門

本書について

適用範囲と目的

AN79953 では、Arm® Cortex®-M0/M0+ベースのプログラム可能なシステムオンチップである PSOC™ 4 についてご紹介します。本資料は、開発者のお客様が PSOC™ 4 アーキテクチャとその開発ツールを探究するために役立つほか、PSOC™ Creator および ModusToolbox™ という PSOC™ 4 用の開発ツールを使用して新規のプロジェクトを作成する際に有用な案内をご提供します。また、PSOC™ 4 に関する深い学習を促進するリソースもご提供します。

対象者

このアプリケーションノートは、PSOC™ および ModusToolbox™ を初めて使用するエンジニア、および組み込みマイクロコントローラーの使用経験があるエンジニアを対象とします。

関連製品ファミリー

すべての [PSOC™ 4 製品](#)

ソフトウェアバージョン

[PSOC™ Creator 4.4 SP2](#) 以降, [ModusToolbox™ 3.2](#) 以降

その他のサンプルコードが必要な場合は、以下のとおり対応いたします。

ModusToolbox™ を使用して増え続ける PSOC™ 4 サンプルコードの一覧にアクセスするには、[GitHub](#) サイトにアクセスしてください。PSOC™ ビデオライブラリは[ここ](#)からご覧ください。

目次

目次

	本書について	1
	目次	2
1	はじめに	4
2	開発エコシステム	5
2.1	PSOC™リソース	5
2.2	ファームウェア/アプリケーション開発	6
2.2.1	ModusToolbox™ ツールパッケージのインストール	6
2.2.2	IDE の選択	6
2.2.3	ModusToolbox™ソフトウェア	8
2.2.4	ModusToolbox™アプリケーション	10
2.2.5	PSOC™ 4 ソフトウェアリソース	11
2.2.5.1	コンフィギュレーター	11
2.2.5.2	PSOC™ 4 ライブラリ マネージメント	12
2.2.5.3	PSOC™ 4 ソフトウェア開発キット	12
2.2.6	ModusToolbox™ヘルプ	13
2.3	他の IDE のサポート	14
2.4	PSOC™ Creator	14
2.4.1	PSOC™ Creator ヘルプ	15
2.5	テクニカル サポート	15
3	PSOC™ 4 の機能セット	16
4	PSOC™が MCU より優れている点	25
4.1	PSOC™ Creator コンポーネントのコンセプト	26
5	はじめての PSOC™ 4 設計	27
5.1	前提条件	27
5.1.1	ハードウェア	27
5.1.2	ソフトウェア	27
5.2	これらの手順の使用	27
5.3	設計について	28
5.4	ボードの BSP の作成	28
5.5	新しいアプリケーションの作成	28
5.6	設計の参照と変更	32
5.6.1	Device configurator を開く	34
5.7	ファームウェアの書き込み	39
5.8	アプリケーションのビルド	44
5.9	デバイスのプログラム	45
5.10	設計のテスト	47
6	PSOC™ Creator を使用するはじめての PSOC™ 4 設計	50

目次

6.1	インストールの前に.....	50
6.1.1	PSOC™ Creator をインストールしましたか?.....	50
6.1.2	開発キットまたは Prototyping Kit をお持ちですか?.....	50
6.1.3	実行中のプロジェクトを参照したいですか?.....	50
6.2	設計について.....	50
6.3	パート 1: 設計の作成.....	51
6.4	パート 2: デバイスのプログラム.....	64
7	まとめ.....	67
	参考資料	68
	改訂履歴	69
	商標	72
	免責事項	73

1 はじめに

1 はじめに

PSOC™ 4 デバイスは、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 Arm® Cortex®-M0 または Cortex®-M0+ 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組込みシステムオンチップです。このようなシステムは、マイクロコントローラー ユニット (MCU) と外付けアナログおよびデジタルペリフェラルの組合せを使用するほとんどのミックスドシグナルの組込みシステムとは異なります。通常、これらのシステムは MCU に加えて、オペアンプ、ADC、Application-specific Integrated Circuit (ASIC) など多くの集積回路が必要です。

PSOC™ 4 は、MCU と外部 IC の組み合わせに代わるものとして低コストなソリューションを提供します。システム全体のコスト削減に加えて、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高い柔軟性、設計のインフィールド チューニングおよび製品化までの時間縮小をご提供します。

PSOC™ 4 の CAPSENSE™ (静電容量タッチ センシング機能) は、比類のない信号対雑音比、クラス最高の耐液性およびボタンやスライダー、トラックパッド、近接センサーなど多種多様なセンサータイプを提供します。PSOC™ 4 は SRAM、プログラマブル ロジックおよび割込みからウェイクアップする機能を保持しながら、クラス最高の低消費電流 150 nA を実現します。また、PSOC™ 4 は非保持の電力モードでウェイクアップ機能を保持し、わずか 20 nA の消費電流です。PSOC™ 4 ファミリー デバイスは、Bluetooth® Low Energy (Bluetooth® LE) 無線システムを搭載する PSOC™ 4 Bluetooth® LE も含んでいます。PSOC™ 4 Bluetooth® LE の詳細については [AN91267](#) を参照してください。

本ドキュメントを使用するにあたって

この後の数ページでは、PSOC™ 4 自身と、PSOC™、ModusToolbox™、および PSOC™ Creator を使用して設計する利点について説明します。または、ModusToolbox™で簡単なデザインをすぐに作成することもできます。[はじめての PSOC™ 4 設計](#)に進んでください。PSOC™ Creator を使用する場合は、[PSOC™ Creator を使用するはじめての PSOC™ 4 設計](#)に進んでください。

2 開発エコシステム

2 開発エコシステム

2.1 PSOC™リソース

インフィニオン web ページで利用できる豊富な情報は、適切な PSOC™ デバイスを選択し、お客様の設計に迅速かつ効果的にデバイスに統合するのを支援します。以下は、PSOC™ 4 リソースの要約です。

- **概要:** [PSOC™ ポートフォリオ](#)
- **製品セレクト:** [PSOC™ 4](#)。さらに、[PSOC™ Creator](#) にはデバイス選択ツールを含みます。
- **データシート**は、各ファミリの電気的仕様を掲載し説明します。
- **アプリケーションノート**は、基本的なレベルから高度なレベルまでの幅広いトピックに触れており、以下のものが含まれます。
 - [AN88619](#): PSOC™ 4 Hardware Design Considerations
 - [AN73854](#): PSOC™ Creator - ブートローダの導入
 - [AN89610](#): PSOC™ Arm® Cortex® コードの最適化
 - [AN86233](#): PSOC™ 4 MCU の低消費電力モードおよび消費電力低減技術
 - [AN57821](#): PSOC™ 3、PSOC™ 4、および PSOC™ 5LP のアナログ/デジタル混在回路基板レイアウトの注意事項
 - [AN89056](#): PSOC™ 4 - IEC 60730 class B and IEC 61508 SIL Safety Software Library
 - [AN64846](#): Getting Started with CAPSENSE™
 - [AN85951](#): PSOC™ 4 および PSOC™ MCU CAPSENSE™ デザインガイド
- **サンプルコード**は、製品の機能と使用法を示します。
- **テクニカルリファレンス マニュアル (TRM):** 各 PSOC™ 4 デバイスファミリのアーキテクチャとレジスタの詳細な説明を掲載します。
- **PSOC™ 4 programming specification** は、PSOC™ 4 不揮発性メモリのプログラムに必要な情報を提供します。
- **開発キット:**
 - [CY8CKIT-040](#), [CY8CKIT-042](#), [CY8CKIT-044](#), [CY8CKIT-046](#), [CY8CKIT-042-BLE](#), [CY8CKIT-045S](#), および [CY8CKIT-041S-MAX](#) PSOC™ 4 Pioneer Kits は使いやすい安価な開発プラットフォームです。これらのキットには、Arduino 準拠シールドおよび Digilent Pmod ドーターカードの専用コネクタを搭載します。
 - [CY8CKIT-043](#), [CY8CKIT-145-40XX](#), [CY8CKIT-147](#), [CY8CKIT-149](#), [CY8CPROTO-040T](#) および [CY8CPROTO-041TP](#) は PSOC™ 4 デバイスをサンプリングするための低コストプロトタイプングプラットフォームです。
 - [CY8CKIT-040T](#) は、PSOC™ 4000T デバイスの低消費電力 CAPSENSE™、低消費電力ウェイクオンタッチおよび耐液体機能を示す低コストの評価キットです。
 - [MiniProg3](#) または [MiniProg4](#) キットは、フラッシュプログラムとデバッグのためのインターフェースを提供します。
 - 統合開発環境 (IDE): PSOC™ 4 を使用したアプリケーション開発に使用できる開発プラットフォームには、[ModusToolbox](#) と [PSOC™ Creator](#) の 2 つがあります。
 - [PSOC™ 4 CAD ライブラリ](#)は、一般的なツールのフットプリントと回路図のサポートを提供します。[IBIS モデル](#)もご利用いただけます。
- **インフィニオン web サイト**のトレーニングビデオは、[PSOC™ 4101 シリーズ](#)を含む幅広いトピックで利用できます。
- **インフィニオンコミュニティ**は、世界中の仲間の PSOC™ 開発者との接続を 24 時間年中無休で可能にし、専用の [PSOC™ 4 MCU コミュニティ](#)をホストします。

2 開発エコシステム

2.2 ファームウェア/アプリケーション開発

PSOC™ 4 を使用したアプリケーション開発に使用できる開発プラットフォームは 2 つあります。

- ModusToolbox™:** このソフトウェアには、構成ツール、低レベルドライバー、ミドルウェアライブラリ、および MCU とワイヤレスアプリケーションの作成を可能にするその他のパッケージが含まれます。すべてのツールは Windows、macOS、Linux 上で動作します。また ModusToolbox™ には、すべての ModusToolbox™ ツールと統合されたフローを提供する Eclipse IDE も含まれます。Visual Studio Code、IAR Embedded Workbench、Arm® MDK (μVision) などの他の IDE もサポートしています。

ModusToolbox™ ソフトウェアは、スタンドアロンのデバイスとミドルウェアのコンフィギュレーターをサポートしています。コンフィギュレーターを使用して、デバイス内のさまざまなブロックの構成を設定し、ファームウェア開発で使用できるコードを生成します。ModusToolbox™ は、すべての PSOC™ 6 MCU および最新の PSOC™ 4 MCU デバイスをサポートします。表 1 に、サポートされている PSOC™ 4 デバイスを示します。インフィニオンは、サポートされている PSOC™ 4 デバイスのすべてのアプリケーション開発には ModusToolbox™ を使用することを推奨します。詳細については、[ModusToolbox™ tools package user guide](#) を参照してください。

表 1 ModusToolbox™ でサポートされる PSOC™ 4 デバイスのリスト

デバイス ¹⁾	ModusToolbox™	PSOC™ Creator
PSOC™ 4000S, PSOC™ 4100S, PSOC™ 4100S Plus, PSOC™ 4100S Plus 256K	有	有
PSOC™ 4100S Max, PSOC™ 4000T, PSOC™ 4100T Plus	有	無
他のすべての PSOC™ 4 シリーズ	無	有

1) PSOC™ 4 に完全な PSOC™ 4 ポートフォリオを設定します。

ライブラリと有効化ソフトウェアは [GitHub](#) で入手できます。

ModusToolbox™ のツールとリソースもコマンドラインで使用できます。詳細については、[ModusToolbox™ tools package user guide](#) の ModusToolbox™ build system セクションを参照してください。

- PSOC™ Creator:** PSOC™ Creator は無料の Windows ベースの IDE です。これにより PSOC™ 3, PSOC™ 4, PSOC™ 5LP, および PSOC™ 6 MCU システムのハードウェアとファームウェアの同時設計が可能になります。アプリケーションは、回路図キャプチャと 150 を超える事前検証済みの本番環境対応の周辺機器コンポーネントを使用して作成されます。

2.2.1 ModusToolbox™ ツールパッケージのインストール

詳細は [ModusToolbox™ tools package installation guide](#) を参照してください。

2.2.2 IDE の選択

図 1 は、適切な IDE を選択するために役立ちます。

2 開発エコシステム

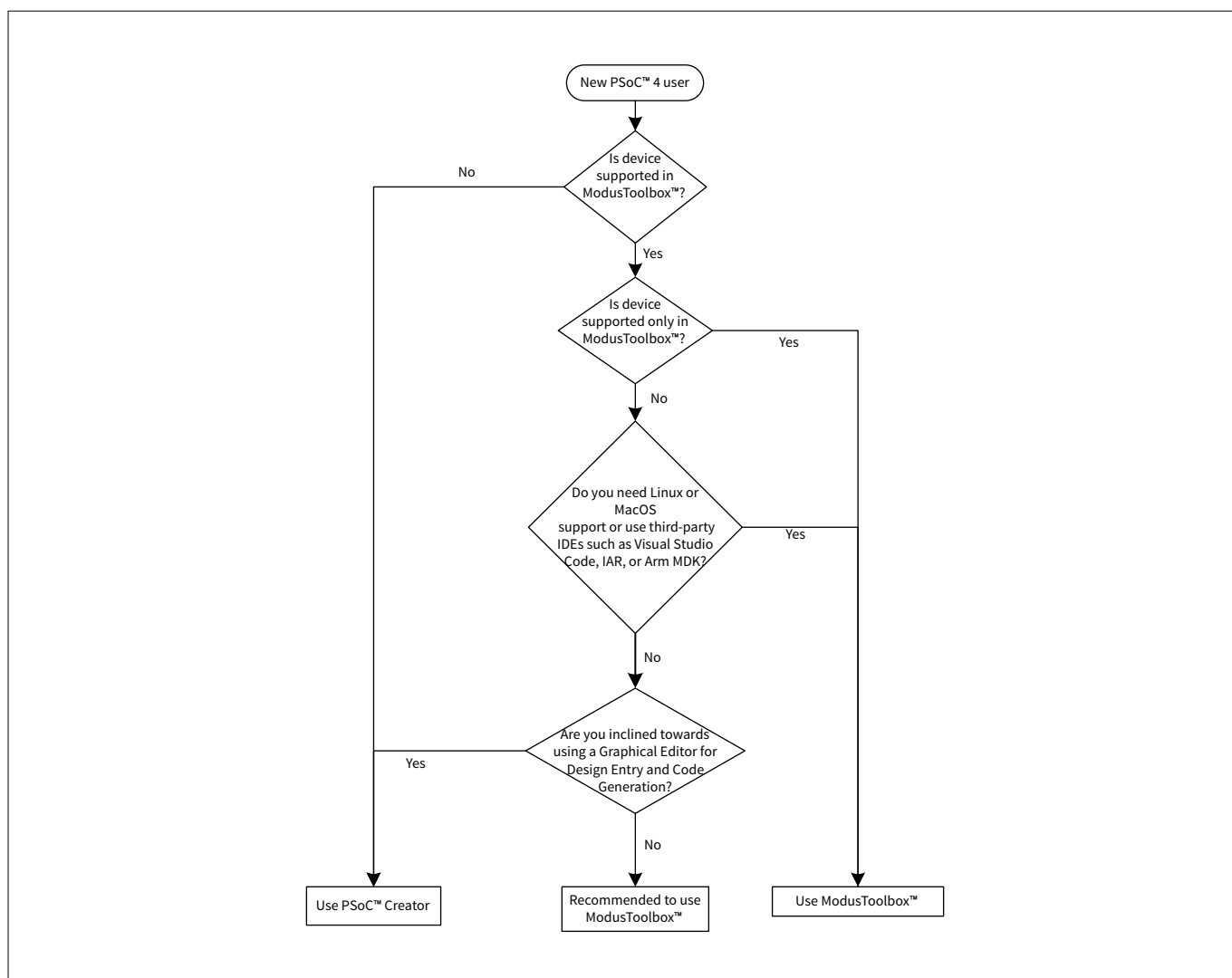


図 1 IDE の選択

最新世代のツールセットである ModusToolbox™ソフトウェアは、Windows、Linux、および macOS プラットフォームに対応します。ModusToolbox™は、Eclipse IDE、Visual Studio Code、Arm® MDK (μVision)、および IAR Embedded Workbench などのサードパーティ IDE にも対応します。ツールパッケージには、サポートするすべての IDE 用の実装が含まれています。ツールは、すべての PSOC™ 4 をサポートしています。また、関連する BSP とライブラリコンフィギュレーターは、3 つのホスト OS すべてで動作します。

2 開発エコシステム

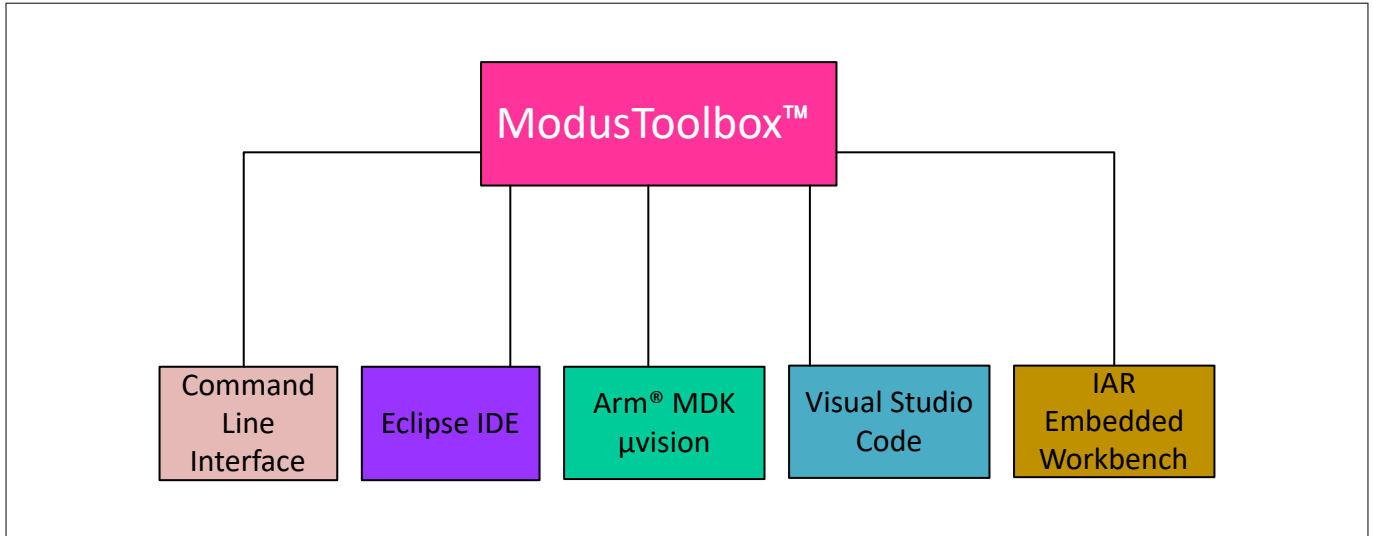


図 2 ModusToolbox™ 環境

UDB (ユニバーサル デジタル ブロック) や USB など、PSOC™ 4 の一部の機能は、ModusToolbox™ バージョン 2.x 以前ではサポートされていません。ModusToolbox の™ 新しいバージョンは、USB ホスト機能をサポートし、真のマルチコアデバッグサポートでユーザーエクスペリエンスを向上させます。

IoT デバイスを使用して IoT アプリケーションを構築する場合、または PSOC™ Creator でサポートされていない PSOC™ 4 を使用している場合も ModusToolbox™ の使用を推奨します。

PSOC™ Creator は、Windows 上でのみ実行される独自の実績のあるツールです。この完成度の高い IDE には、コンポーネントの補助で回路図ベースの設計入力をサポートするグラフィカル エディタが含まれます。PSOC™ Creator は、すべての PSOC™ 3、PSOC™ 4、PSOC™ 5LP デバイス、および PSOC™ 4 デバイスのサブセットをサポートします。

設計入力とコード生成にグラフィカルエディタを使用する傾向がある場合、および使用する予定の MCU が IDE でサポートされている場合、または PSOC™ MCU で UDB を使用する場合は、PSOC™ Creator を選択してください。

2.2.3 ModusToolbox™ ソフトウェア

ModusToolbox™ は、統合された MCU およびワイヤレスシステムを作成するための没入型開発経験を可能にし、インフィニオンデバイスを既存の開発方式に統合できるツールとソフトウェアのセットです。これらには、コンフィギュレーション ツール、低レベルドライバ、ライブラリ、オペレーティング システム サポートが含まれ、そのほとんどが Linux、macOS、Windows ホスト環境と互換性があります。

図 3 は、ModusToolbox™ ソフトウェアの一部として利用可能なものを概説します。ModusToolbox™ ソフトウェアの詳細な概要は「[ModusToolbox™ tools package user guide](#)」を参照してください。

2 開発エコシステム

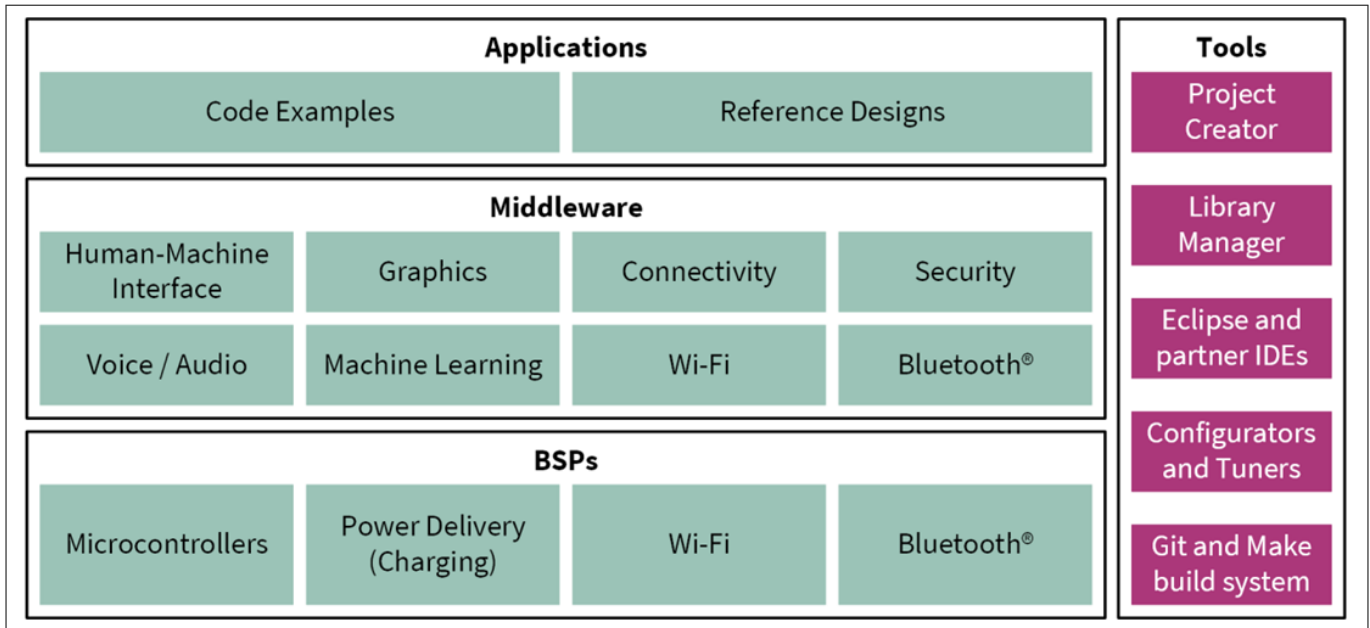


図 3 ModusToolbox™ソフトウェア

ModusToolbox™ ツール パッケージ インストーラーには、デザインコンフィギュレーターとツール、およびビルドシステムインフラストラクチャが含まれます。

ビルドシステムインフラストラクチャには、Eclipse IDE、メイクインフラストラクチャ、およびその他のツールから独立して実行できる新しいプロジェクト作成ウィザードが含まれます。これにより、業界をリードする CAPSENSE™ (ヒューマン マシン インターフェイス)、AIROC™ Wi-Fi および Bluetooth®、セキュリティ、その他さまざまな機能へのアクセスや使い勝手を損なうことなく、コンパイラ、IDE、RTOS、エコシステムを選択できます。

ModusToolbox™ エコシステムの一部であるランタイム ソフトウェアは、コネクティビティコンボ デバイスを使用した Wi-Fi および Bluetooth® アプリケーションの迅速な開発を支援します。詳細については、[ModusToolbox™ run-time software reference guide](#) を参照してください。

デザイン コンフィギュレーターは、BSP/ミドルウェアの設定可能なコード作成を支援するツールです。詳細については、[コンフィギュレーター](#)に進んでください。

すべてのアプリケーションレベルでの開発フローは、提供する低レベルリソースに依存します。これらには以下が含まれます。

- **ボード サポート パッケージ (BSP):** BSP は、ボード固有のドライバおよびその他の機能を含むファームウェアのレイヤです。BSP は、ボードを初期化し、ボードレベルのペリフェラルにアクセスできる API のライブラリセットです。これには、PSOC™ 4 用の Peripheral driver library (PDL) などの低レベルリソースが含まれ、ボードペリフェラル用のマクロがあります。カスタム BSP を作成して、エンドアプリケーションボードのサポートを有効にできます。カスタム BSP を作成するには、[BSP Assistant](#) を参照してください。
- **PSOC™ 4 ペリフェラルドライブライブラリ (PDL):** PDL は、デバイス ヘッド ファイル、スタートアップコード、ペリフェラルドライバを単一のパッケージに統合します。PDL は PSOC™ 4 ファミリーをサポートします。ドライバは、ハードウェア機能を一連の使いやすい API に抽象化します。これらは、PDL API リファレンスに完全に文書化されています。
PDL により、レジスタの使用法とビット構造を理解する必要性が減るため、PSOC™ 4 シリーズの広範なペリフェラル セットのソフトウェア開発が容易になります。アプリケーションのドライバを構成してから、API 呼び出しを使用してペリフェラルを初期化して使用します。
- **ミドルウェア (MW):** アプリケーションに特定の機能を提供する広範なミドルウェアライブラリ。[利用可能なミドルウェア](#)は、コネクティビティ (OTA、Bluetooth®、AWS IoT、Bluetooth® LE、Secure Sockets) から、PSOC™ 4 に特化した機能 (CAPSENSE™、USB、デバイスのファームウェア アップグレード (DFU)、emWin) まで多岐にわたります。ミドルウェアはすべて、GitHub リポジトリを介してライブラリとして提供されます。

2 開発エコシステム

2.2.4 ModusToolbox™アプリケーション

ModusToolbox™ v3.x のリリースに伴い、マルチコアのサポートが導入され、以前のバージョンの ModusToolbox™ からフォルダ構造が若干変更されました。ただし、PSOC™ 4 はシングルコアのみです。

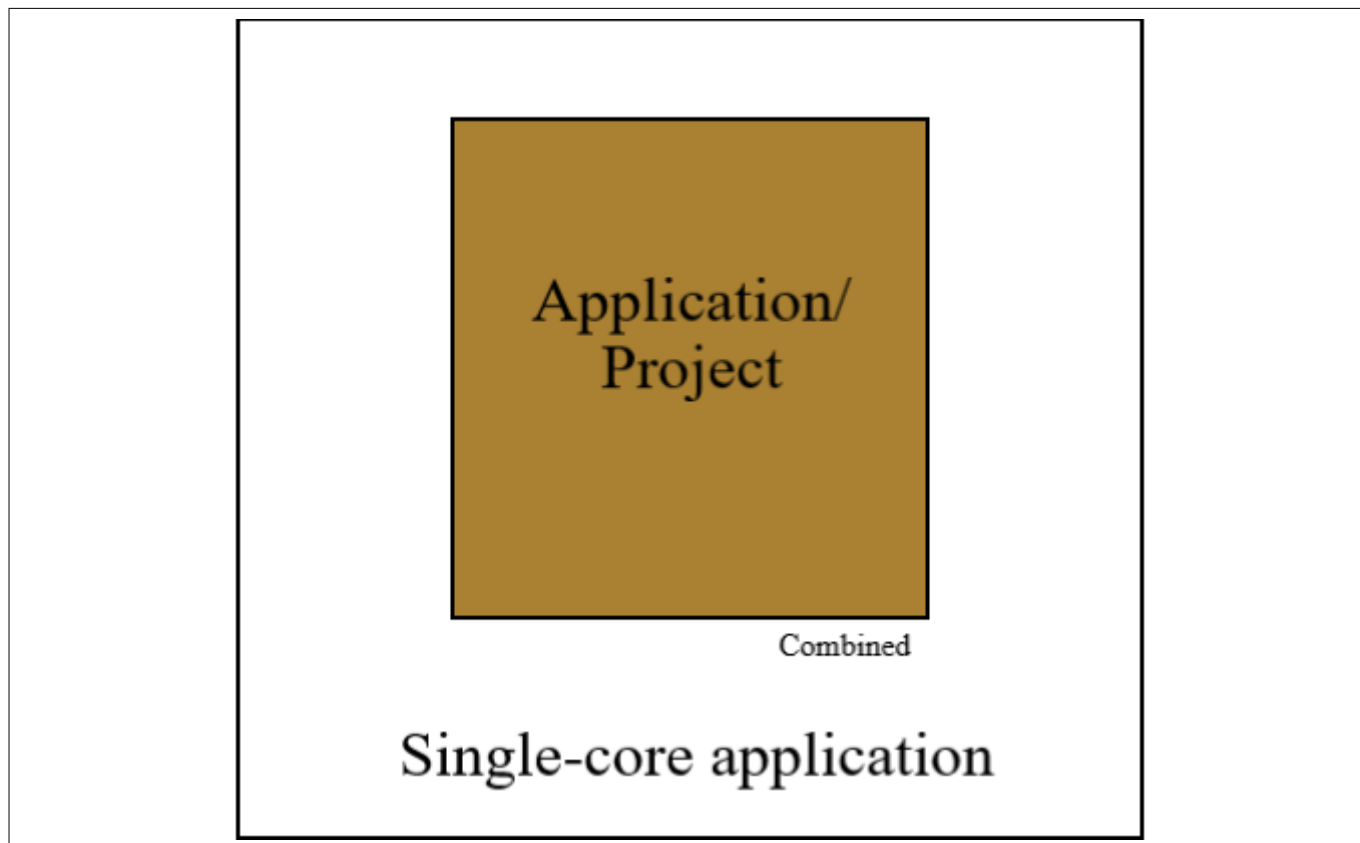


図 4 アプリケーションの種類

シングルコアアプリケーションの例の新しいフォルダ構造を以下に示します。

2 開発エコシステム

```

<root>
  ApplicationName
  ->Makefile (MTB_TYPE=COMBINED)
  ->deps
    lib1.mtb (local)
    lib2.mtb (shared)
  ->libs
    lib1 (Infineon Git repo)
  ->bsps
    TARGET_BSP1 (not an Infineon Git repo; completely app-owned)
  ->templates
    TARGET_BSP1
      design.modus
      design.capsense
  ->main.c
  ->helper.h
  ->helper.c
  mtb_shared
    lib2/... (Infineon Git repo)

```

図 5 シングルコアアプリケーションのフォルダ構造

2.2.5 PSOC™ 4 ソフトウェアリソース

PSOC™ 4 のソフトウェアには、コンフィギュレーター、ドライバ、ライブラリ、ミドルウェアのほか、各種ユーティリティ、Makefile、スクリプトが含まれます。また、関連ドライバ、ミドルウェアや、IoT デバイス、接続ソリューションで使用するサンプルコードも含まれます。どのツールでも、お好みの環境で使用できます。

2.2.5.1 コンフィギュレーター

ModusToolbox™は、ハードウェアブロックの設定を容易にするコンフィギュレーターと呼ばれるグラフィカルアプリケーションを提供します。例えば、全てのドキュメントを検索してシリアル通信ブロックを目的の設定の UART として設定する代わりに、適切なコンフィギュレーターを開き、ボーレート、パリティ、およびストップビットを設定します。ハードウェア構成を保存すると、ツールは「C」コードを生成して、ハードウェアを目的の設定で初期化します。MCU ハードウェアに固有の項目を設定する BSP コンフィギュレーターと、ミドルウェアライブラリのオプションを設定するライブラリコンフィギュレータの 2 種類があります。

コンフィギュレーターは互いに独立していますが、一緒に使用して柔軟な構成オプションを提供できます。これらは、スタンドアロンで、または他のツールと組み合わせて、または完全な IDE 内で使用できます。コンフィギュレーターは次の目的で使用されます。

- オプションを設定し、コードを生成して、ドライバを構成する
- ペリフェラルの端子やクロックなどの接続を設定する
- オプションを設定し、コードを生成して、ミドルウェアを構成する

PSOC™ 4 アプリケーションの場合、使用可能なコンフィギュレーターは次のとおりです。

- Device Configurator: システム (プラットフォーム) 機能と基本的な周辺機器 (UART、タイマー、PWM など) をセットアップします。

2 開発エコシステム

- CAPSENSE™ Configurator および Tuner: CAPSENSE™を構成し、必要なコードを生成します。
- Smart I/O Configurator: スマート I/O を構成します。

上記の各コンフィギュレーターは独自のファイルを作成します (例えば、CAPSENSE™の場合は design.cycapsense)。BSP コンフィギュレーター ファイル (例えば、design.modus や design.cycapsense) はデフォルトコンフィギュレーションで BSP の一部として提供され、ライブラリー コンフィギュレーター (例えば、design.cybt) はアプリケーションによって提供されます。アプリケーションがインフィニオン BSP に基づいて作成される場合、アプリケーションはインフィニオン BSP リポジトリの BSP コンフィギュレーター ファイルを使用します。ModusToolbox™ ソフトウェアを使用して、アプリケーションの要件に応じてすべてのコンフィギュレーター ファイルをカスタマイズ/作成できます。カスタム BSP を作成するには、[BSP Assistant](#) を参照してください。詳細については、[ModusToolbox™ヘルプ](#)を参照してください。

2.2.5.2 PSOC™ 4 ライブラリ マネージメント

アプリケーションは、プロジェクト用の共有/ローカル ライブラリを持つことができます。異なるプロジェクトで同じライブラリの異なるバージョンを使うこともできます。共有ライブラリは mtb_shared ディレクトリにダウンロードされます。アプリケーションは deps フォルダを使って、ライブラリの依存関係を追加します。deps フォルダには、ModusToolbox™が git リポジトリをダウンロードするために使用する、拡張子 .mtb のファイルが含まれます。これらのライブラリは ModusToolbox™ プロジェクトに直接依存しています。

ライブラリ マネージャは、プロジェクトのライブラリを追加/削除/更新できます。また、GitHub で利用可能なマニフェスト リポジトリを使用して、特定のライブラリが他のライブラリに直接依存しているかどうかを識別し、その依存関係をすべて取得します。これらに対応したライブラリは ModusToolbox™ プロジェクトに間接的に依存しています。これらの依存関係は、libs フォルダの下で確認できます。詳細については、<install_dir> /ModusToolbox/ tools_<version>/library-manager/docs/library-manager.pdf の [Library Manager user guide](#) を参照してください。

2.2.5.3 PSOC™ 4 ソフトウェア開発キット

ModusToolbox™ エコシステムは、PSOC™ 4 のソフトウェア開発を可能にする重要なソースコードとツールを提供します。

- ハードウェアの設定方法を指定します。
- そのためにコードを生成し、ファームウェアで使用してください。
- Bluetooth® LE 接続や FreeRTOS のような追加機能のための様々なミドルウェアライブラリを含みます。

このソースコードにより、サポートされているデバイスのファームウェアの開発が容易になります。レジスタセットを理解しなくても、ファームウェアをすばやくカスタマイズしてビルドするために役立ちます。

ModusToolbox™環境では、コンフィギュレーターを使用して、デバイス、または Bluetooth® LE スタックまたは CAPSENSE™機能などのミドルウェアライブラリを構成できます。BSP コンフィギュレーター ファイルは、ペリフェラルドライバライブラリコードを使用して、デバイスのペリフェラル、ピン、メモリを設定します。ミドルウェアは、複数のプラットフォームで使用できるように、各機能ごとに個別のライブラリとして提供されます。例えば、abstractionrtos、lwip、usb など。

レジスタレベルでのファームウェア開発者であっても、PDL からのドライバソースコードを参照する必要があります。PDL には、プロジェクトに必要なすべてのデバイス固有のヘッダーファイルとスタートアップコードが含まれます。また、各ドライバーのリファレンスとしても機能します。PDL はソースコードとして提供されるため、レジスタレベルでハードウェアにアクセスする方法を確認できます。

一部のデバイスは特定の周辺機器をサポートしていません。PDL は、サポートされているデバイスのすべてのドライバーのスーパーセットです。このスーパーセットの設計は、以下のことを意味します。

- ペリフェラルの初期化、構成、および使用に必要なすべての API 要素が利用可能です。
- PDL は、使用可能なペリフェラルに関係なく、さまざま PSOC™ 4 で有効です。
- PDL は対象となるペリフェラルが選択されたデバイスに存在することを保証するためのエラー チェックを含んでいます。

2 開発エコシステム

これにより、周辺機器が利用可能である限り、コードは PSOC™ 4 ファミリー全体で互換性を維持できます。デバイスヘッダーファイルは、デバイスで使用可能な周辺機器を指定します。コードがサポートされていないペリフェラルを使用しようとすると、コンパイル中にエラーが発生します。ペリフェラルを使用するコードを書く前に、そのペリフェラルのサポートを確認するため、そのデバイスのデータシートを参照してください。

下図に示すように、ModusToolbox™ソフトウェアを使用すると、以下のことができます。

1. BSP (Project Creator) を選択してください。
2. 各アプリケーションがサポートする BSP によってフィルタリングされたスターターアプリケーションのリストに基づいて、新しいアプリケーションを作成します (Project Creator)。
3. BSP またはミドルウェアライブラリを追加してください (ライブラリマネージャ)。
4. PSOC™ 4 用のまたは PDL を使用してアプリケーションファームウェアを開発してください (任意の IDE またはコマンドライン)。

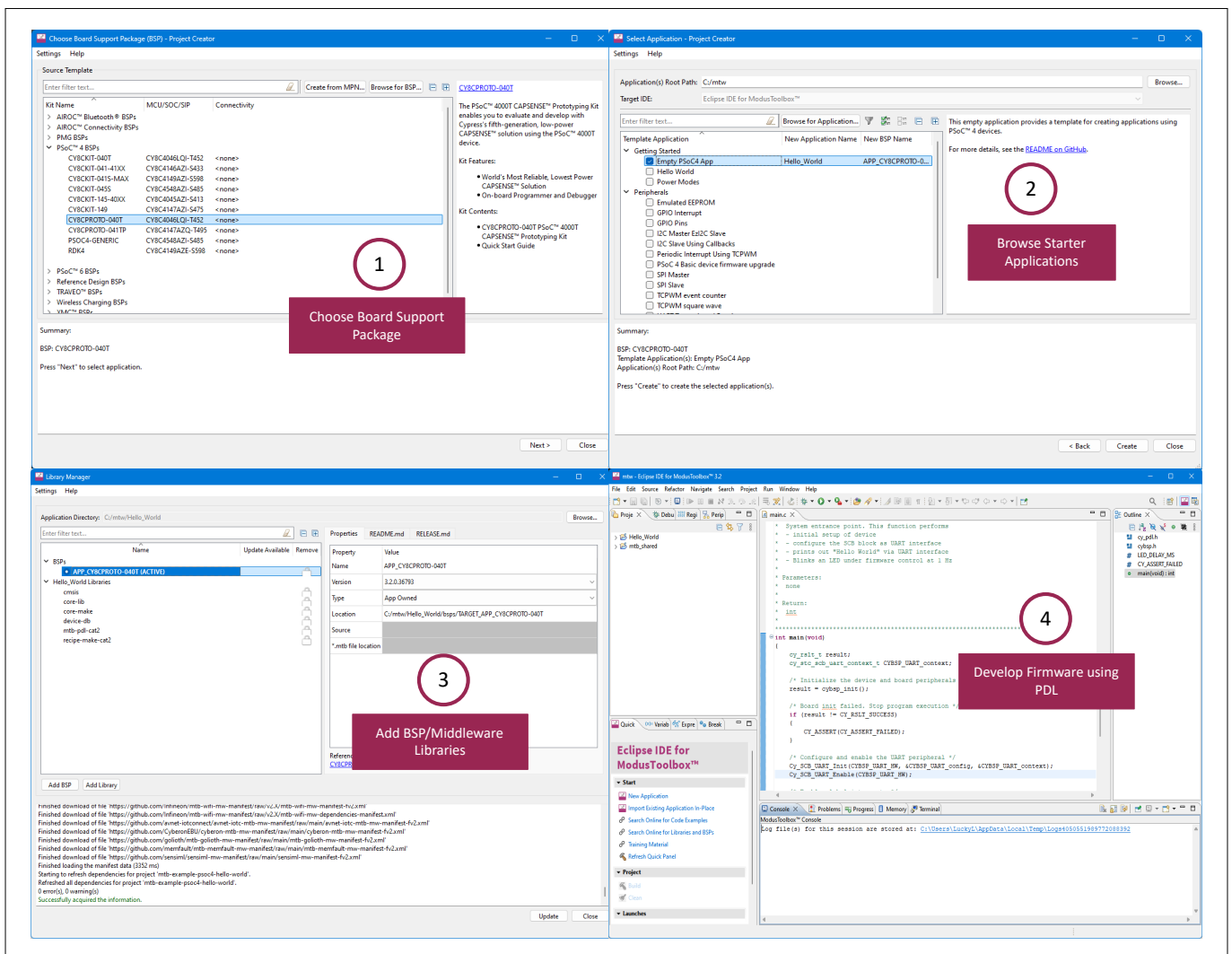


図 6 ModusToolbox™のリソースとミドルウェア

2.2.6 ModusToolbox™ヘルプ

ModusToolbox™ エコシステムは、ドキュメントとトレーニングを提供します。ModusToolbox™ ソフトウェアの Eclipse IDE を起動し、以下の Help メニューに移動することでアクセスできます。

Help > ModusToolbox™ General Documentation を選択してください。

- **ModusToolbox™ Documentation Index:** ModusToolbox™ソフトウェアの一部として含まれるさまざまなタイプのドキュメントへの簡単な説明とリンクを提供します。

2 開発エコシステム

- **ModusToolbox™ Installation Guide:** ModusToolbox™ソフトウェアをインストールするための手順を提供します。
- **ModusToolbox™ User Guide:** このガイドでは主に、アプリケーションの構築、プログラミング、デバッグの ModusToolbox™の側面を説明します。また、IDE とともにインストールされるツールのさまざまな側面についても説明します。
- **ModusToolbox™ Training Material:** トレーニング資料へのリンク: <https://github.com/Infineon/training-modustoolbox>
- **Release Notes**

ModusToolbox™の Eclipse IDE に関する資料については、**Help > Eclipse IDE for ModusToolbox™ Documentation** を選択してください。

- **User Guide:** Eclipse IDE を使用したアプリケーションの作成、ビルド、プログラミング、およびデバッグに関する説明を提供します。
- **Eclipse IDE Survival Guide**

2.3 他の IDE のサポート

Eclipse IDE、[IAR Embedded Workbench](#)、[Keil μVision 5](#)、[Visual Studio Code](#) など、お好みの IDE を使用して PSOC™ 4 のファームウェアを開発できます。

ModusToolbox™ コンフィギュレーターは、Eclipse IDE を使用せずに、PSOC™ 4 リソースやその他のミドルウェアコンポーネントのセットアップと設定に使用できるスタンドアロンツールです。デバイス コンフィギュレーターとミドルウェア コンフィギュレーターは、アプリケーション ワークスペース内の `design.x` ファイルを使用します。生成されたソースコードを指定して、IDE でファームウェアの開発を続けることができます。

デバイス設定に変更がある場合は、コンフィギュレーターを使用して `design.x` ファイルを編集し、コードを再生成してください。ModusToolbox™ソフトウェアに付属のコンフィギュレーション ツールを使用してリソース コンフィギュレーションを生成することを推奨します。

詳細については、[ModusToolbox™ tools package user guide](#) を参照してください。

2.4 PSOC™ Creator

[PSOC™ Creator](#) は、PSOC™システムのハードウェアとファームウェアの同時編集、コンパイル、およびデバッグを可能にする IDE です。[図 7](#) に示すように、PSOC™ Creator を閲覧するには、次の操作を行ってください。

1. コンポーネントをドラッグ&ドロップして、ハードウェア システム設計の構築
2. アプリケーションのファームウェアと PSOC™ハードウェアの共同設計
3. コンフィギュレーション ツールを使ったコンポーネントの構成
4. 100 以上のコンポーネントを含むライブラリの利用
5. コンポーネント データシートの閲覧

2 開発エコシステム

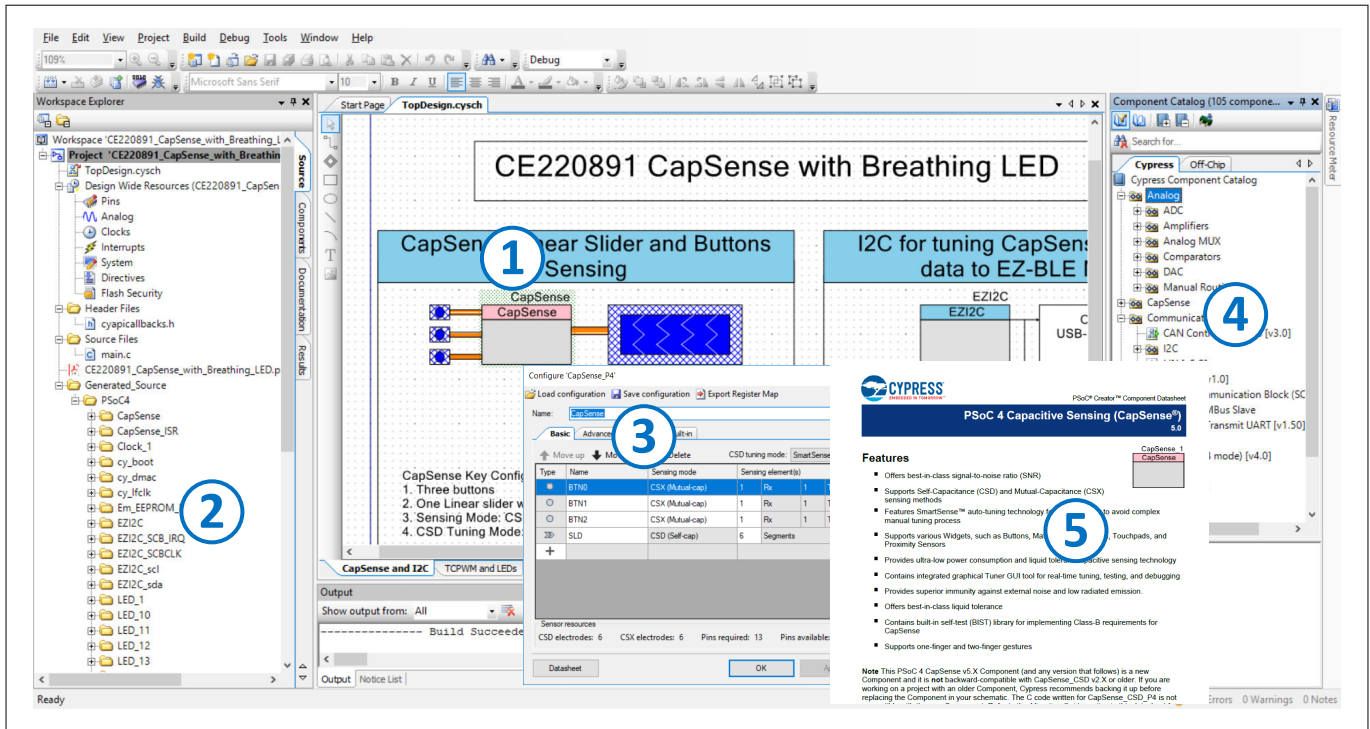


図 7 PSOC™ Creator の特長

2.4.1 PSOC™ Creator ヘルプ

PSOC™ Creator のホームページから PSOC™ Creator の最新版をダウンロードし、PSOC™ Creator を起動し、以下の手順に進んでください。

- Quick Start Guide: Help > Documentation > Quick Start Guide を選択してください。このガイドは PSOC™ Creator プロジェクトを開発するための基礎を提供します。
- Simple Component example project: File > Open > Example projects を選択してください。これらのサンプルプロジェクトは PSOC™ Creator コンポーネントの設定と使用方法を示します。
- System Reference Guide: Help > System Reference > System Reference Guide を選択してください。このガイドは PSOC™ Creator により提供されるシステム機能を一覧で説明します。
- Component datasheets: コンポーネントを右クリックして Open Datasheet を選択してください。すべての PSOC™ 4 コンポーネント データシートの一覧は、[PSOC™ 4 Component datasheets](#) ページに掲載されています。
- [PSOC™ Creator training videos](#): これらの動画は PSOC™ Creator 入門のステップ バイ ステップ方法を提供します。
- Document Manager: PSOC™ Creator が提供するドキュメント マネージャーにより、容易にドキュメントリソースを検索し、確認できます。ドキュメント マネージャーを開くためには、メニューアイテムの Help > Document Manager を選択します。

2.5 テクニカル サポート

ご質問については弊社のテクニカル サポート チームが対応します。[テクニカルサポート](#) ページでサポートリクエストを作成できます。サポートチームは、[GitHub](#) リポジトリに投稿された質問、問題、バグレポートを確認して対応します。

3 PSOC™ 4 の機能セット

3 PSOC™ 4 の機能セット

PSOC™ 4 は図 8 に示すように、CPU とメモリ サブシステム、デジタル サブシステム、アナログ サブシステム、システム リソースを含む幅広い機能を備えます。次の節は各機能について説明します。詳細については、PSOC™ 4 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™ リソース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。

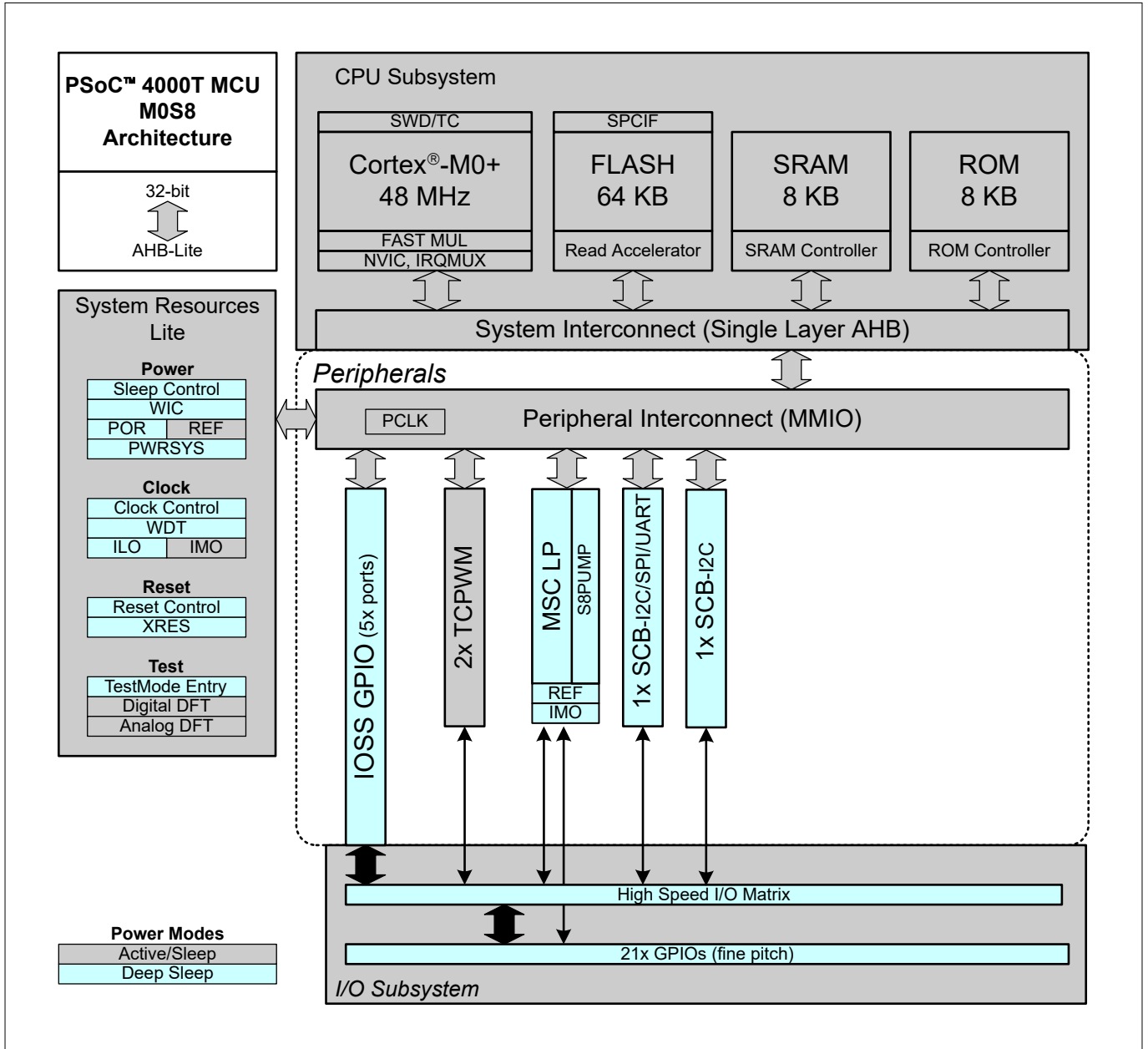


図 8 PSOC™ 4000T アーキテクチャ

PSOC™ 4 ポートフォリオは、Arm® CM0 および CM0+マイクロコントローラーのいくつかのファミリで構成されます。ポートフォリオ内のほとんどのデバイスには、静電容量センシングアプリケーション用の CAPSENSE™テクノロジーが搭載されます。PSOC™ 4 ポートフォリオの他の主要な機能には、プログラム可能なアナログブロックを介したカスタマイズ可能なアナログフロントエンドと、USB、コントローラーエリアネットワーク (CAN)、Bluetooth® LE などの有線および無線接続オプションが含まれます。これらの独自の機能により、PSOC™ 4 は業界で最も柔軟でスケラブルな低電力ミックスドシングナルアーキテクチャを実現しました。PSOC™ 4 デバイスは、表 2 に示すように、さまざまな機能に基づいてさまざまなファミリに分類されます。

3 PSOC™ 4 の機能セット

表 2 PSOC™ 4 ファミリ

分類	ファミリ	特長	詳細
入門レベル	PSOC™ 4000 ファミリ	CAPSENSE™	表 3
インテリジェントアナログ	PSOC™ 4100 ファミリ	CAPSENSE™ + プログラム可能なアナログ	表 4
プログラム可能なデジタル	PSOC™ 4200 ファミリ	CAPSENSE™ + プログラム可能なアナログ + プログラム可能なデジタルブロック	表 5
特定用途向け	PSOC™ 4500 ファミリ	CAPSENSE™ + モーター制御	表 6
	PSOC™ 4700 ファミリ	CAPSENSE™ + Inductive Sensing	
アナログコプロセッサ ¹⁾	PSOC™ 4A00 ファミリ	CAPSENSE™ + プログラム可能なアナログブロック	AN211293

1) PSOC™ 4 アナログコプロセッサファミリデバイスの使用を開始するためには、[AN211293](#) を参照してください。

注: 表 3 および表 4 で、緑色で強調表示されている列のデバイスファミリは ModusToolbox™ でサポートされていることを示します。

表 3 PSOC™ 4000 ファミリの特長

特長	PSOC™ 4000	PSOC™ 4000S	PSOC™ 4000T	
CPU	16 MHz Cortex® -M0	48 MHz Cortex® -M0+	48 MHz Cortex® -M0+	
フラッシュメモリ	16 KB	32 KB	64 KB	
SRAM	2 KB	4 KB	8 KB	
GPIO	20	36	21	
CAPSENSE™	センスピン	16	35	19
	タッチでの超低消費電力ウェイク(WoT)	無	無	有
	CAPSENSE™ IP 世代	第 4 世代	第 5 世代	第 5 世代 低消費電力
シングルスロープ ADC (10 ビット 46 ksps)	なし	1	なし	
コンパレータ	固定閾値の 1 つの CSD コンパレータ (1.2 V)	ウェイクアップ機能付き 2 つの低消費電力コンパレータ	なし	
IDAC ¹⁾	1 つの 7 ビットと 1 つの 8 ビット	2 つの 7 ビット	なし	
スマート I/O ポート	なし	2	なし	
電源電圧範囲	1.71 V ~ 5.5 V	1.71 V ~ 5.5 V	1.71 V ~ 5.5 V	

(続く)

3 PSOC™ 4 の機能セット

表 3 (続き) PSOC™ 4000 ファミリの特長

特長	PSOC™ 4000	PSOC™ 4000S	PSOC™ 4000T
低消費電力モード	2.5 μA でのディープスリープ	2.5 μA でのディープスリープ	2.5 μA でのディープスリープ
セグメント LCD 駆動	なし	4 COM セグメント LCD 駆動	なし
シリアル通信	1 個の I2C	プログラム可能な I2C, SPI, または UART を備えた 2 つの SCB	プログラム可能な I2C, SPI, または UART を備えた 1 つの SCB。I ² C のみを備えた 1 つの SCB
タイマーカウンターパルス幅変調器 (TCPWM)	1	5	2
クロック	内部メイン発振器 (IMO)	24 MHz/32 MHz	24 MHz ~ 48 MHz
	内部低速発振器 (ILO)	32 kHz 内部 ILO	40 kHz
	時計水晶発振器 (WCO)	なし	32 kHz
電源監視	パワーオンリセット (POR), 電圧低下検出 (BOD)	POR, BOD	POR, BOD
対応するキット	CY8CKIT-040 pioneer kit	-	CY8CKIT-040T PSOC™ 4000T CAPSENSE™ Evaluation Kit および CY8CPROTO-040T PSOC™ 4000T CAPSENSE™ Prototyping Kit
対応する IDE	PSOC™ Creator	PSOC™ Creator, ModusToolbox™	ModusToolbox™

1) IDAC は、CAPSENSE™を使用しない場合にのみ利用可能です。詳細は、各 PSOC™ 4 アーキテクチャ TRM を参照してください。

3 PSOC™ 4 の機能セット

表 4 PSOC™ 4100 ファミリの特長

特長	PSOC™ 4100	PSOC™ 4100S	PSOC™ 4100S Plus	PSOC™ 4100PS	PSOC™ 4100M	PSOC™ 4100 BL ⁽¹⁾	PSOC™ 4100S Max	PSOC™ 4100T Plus
CPU	24 MHz Cortex®-M0	48 MHz Cortex®-M0+	48 MHz Cortex®-M0+	48 MHz Cortex®-M0+	24 MHz Cortex®-M0	24 MHz Cortex®-M0	48 MHz Cortex®-M0+	48 MHz Cortex®-M0+
DMA	該当なし	該当なし	8 チヤネル	8 チヤネル	8 チヤネル	8 チヤネル	16 チヤネル	8 チヤネル
フラッシュメモリ	32 KB	64 KB	128 KB	32 KB	128 KB	256 KB	384 KB	128 KB
SRAM	4 KB	8 KB	16 KB	4 KB	16 KB	32 KB	32 KB	32 KB
GPIO	36	36	54	38	55	36	84	53
CAPSENSE™	1 チヤネル, 35 センサー	1 チヤネル, 53 センサー	1 チヤネル, 53 センサー	1 チヤネル, 33 センサー	2 チヤネル, 54 センサー	1 チヤネル, 35 センサー	2 チヤネル, 80 センサー (32 control mux)	1 チヤネル, 32 センサー
シーケンサ付き 12 ビット SAR ADC	806 KSPS	1 MSPS	1 MSPS	1 MSPS	806 KSPS	806 KSPS	1 MSPS	1 MSPS
オペアンプ (プログラム可能)	2	2	2	4/PGA	4	2	2	なし
プログラム可能電圧リファレンス (PVref)	なし	なし	なし	4 チヤネル	なし	なし	なし	なし
電圧 DAC (VDAC)	なし	なし	なし	2つの 13 ビット VDAC	なし	なし	なし	なし
コンパレータ (ウェイクアップ機能付き 低電力)	2	2	2	2	2	2	2	なし

(続く)

3 PSOC™ 4 の機能セット

表 4 (続き) PSOC™ 4100 ファミリの特長

特長	PSOC™ 4100	PSOC™ 4100S	PSOC™ 4100S Plus	PSOC™ 4100S Plus 256K	PSOC™ 4100PS	PSOC™ 4100M	PSOC™ 4100 BL ⁽¹⁾	PSOC™ 4100S Max	PSOC™ 4100T Plus
IDAC ⁽²⁾	1つの7ビットと1つの8ビット	2つの7ビット	2つの7ビット	2つの7ビット	2つの7ビット	2つの7ビットと2つの8ビット	1つの7ビットと1つの8ビット	なし	なし
スマートI/Oポート	なし	2	3	2	1	なし	なし	3	1
電源電圧範囲	1.71V ~ 5.5V	1.71V ~ 5.5V	1.71V ~ 5.5V	1.71V ~ 5.5V	1.71V ~ 5.5V	1.71V ~ 5.5V	1.71V ~ 5.5V	1.71V ~ 5.5V	1.71V ~ 5.5V
低消費電力モード	1.3 μA	2.5 μA	2.5 μA	2.5 μA	2.5 μA	1.35 μA	2.5 μA	2.5 μA	6 μA
	150 nA	該当なし	該当なし	該当なし	該当なし	150 nA	該当なし	該当なし	なし
ストップ	20 nA	該当なし	該当なし	該当なし	該当なし	35 nA	該当なし	該当なし	なし
セグメント LCD 駆動	4 COM	4 COM	4 COM	4 COM	4 COM	4 COM	4 COM	4 COM	なし
プログラム可能なI2C, SPI, またはUARTを備えたSCB	2	3	5	5	3	4	2	5	5 (2 SCB, 3 UART)
TCPWM	4	5	8	8	8	8	4	8	6
CAN	なし	なし	1	なし	なし	なし	なし	1	なし
Bluetooth® LE	なし	なし	なし	なし	なし	なし	4.1/4.2	なし	なし
クロック	3 MHz ~ 24 MHz	24 MHz ~ 48 MHz	24 MHz ~ 48 MHz	24 MHz ~ 48 MHz	24 MHz ~ 48 MHz	3 MHz ~ 48 MHz	24 MHz ~ 48 MHz	24 MHz ~ 48 MHz	24 MHz ~ 48 MHz
ILO	32 kHz	40 kHz	40 kHz	40 kHz	40 kHz	32 kHz	40 kHz	40 kHz	40 kHz

(続く)

3 PSOC™ 4 の機能セット

表 4 (続き) PSOC™ 4100 ファミリの特長

特長	PSOC™ 4100	PSOC™ 4100S	PSOC™ 4100S Plus	PSOC™ 4100PS	PSOC™ 4100M	PSOC™ 4100 BL ¹⁾	PSOC™ 4100S Max	PSOC™ 4100T Plus
電源監視	Nil	32 kHz	32 kHz	32 kHz	32 kHz	32 kHz	32 kHz	32 kHz
	POR, BOD, 低電圧検知 (LVD)	POR, BOD	POR, BOD	POR, BOD	POR, BOD, LVD	POR, BOD, LVD	POR, BOD	POR, BOD
対応するキット	-	-	CY8CKIT-149 prototyping kit	CY8CKIT-147 prototyping kit	CY8CKIT-044 pioneer kit	CY8CKIT-042 BLE pioneer kit	CY8CKIT-041S -MAX pioneer kit	PSOC™ 4100T Plus CAPSENSE™ Prototyping kit
対応する IDE	PSOC™ Creator	PSOC™ Creator, ModusToolbo X™	PSOC™ Creator, ModusToolbo X™	PSOC™ Creator	PSOC™ Creator	PSOC™ Creator	ModusToolbo X™	ModusToolbo X™

1) PSOC™ 4 Bluetooth® LE ファミリデバイスの使用を開始するためには、AN91267 を参照してください。

2) IDAC は、CAPSENSE™を使用しない場合にのみ利用可能です。詳細は、各 PSOC™ 4 アーキテクチャ TRM を参照してください。

3 PSOC™ 4 の機能セット

表 5 PSOC™ 4200 ファミリの特長

特長	PSOC™ 4200	PSOC™ 4200DS	PSOC™ 4200M	PSOC™ 4200L	PSOC™ 4200 BL ¹⁾	
CM0 CPU	48 MHz Cortex® - M0	48 MHz Cortex® - M0	48 MHz Cortex® - M0	48 MHz Cortex® - M0	48 MHz Cortex® - M0	
DMA	なし	8 チャンネル	8 チャンネル	32 チャンネル	なし	
フラッシュメモリ	32 KB	64 KB	128 KB	256 KB	256 KB	
SRAM	4 KB	8 KB	16 KB	32 KB	32 KB	
GPIO	36	21	55	96	36	
CAPSENSE™	1 チャンネル, 35 センサー	なし	2 チャンネル, 54 センサー	2 チャンネル, 94 センサー	1 チャンネル, 35 センサー	
ADC (12 ビット, シーケンサ付き 1 MSPS SAR ADC)	1	なし	1	1	1	
オペアンプ (プログラム可能)	2	なし	2	4	2	
コンパレータ (ウェイクアップ機能付き低電力)	2	2	2	2	2	
IDAC ²⁾	1 つの 7 ビットと 1 つの 8 ビット	なし	2 つの 7 ビットと 2 つの 8 ビット	2 つの 7 ビットと 2 つの 8 ビット	1 つの 7 ビットと 1 つの 8 ビット	
プログラム可能論理ブロック (UDB)	4	4	4	8	4	
スマート I/O ポート	なし	1	なし	なし	なし	
電源電圧範囲	1.71 V ~ 5.5 V	1.71 V ~ 5.5 V	1.71 V ~ 5.5 V	1.71 V ~ 5.5 V	1.71 V ~ 5.5 V	
低消費電力モード	ディープスリープ	1.3 µA	2 µA	1.3 µA	1.3 µA	1.5 µA
	ハイバネート	150 nA	該当なし	150 nA	150 nA	150 nA
	ストップ	20 nA	該当なし	20 nA	20 nA	20 nA
セグメント LCD 駆動	4 COM	なし	4 COM	8 COM	4 COM	
プログラム可能な I2C, SPI, または UART を備えた SCB	2	3	4	4	2	
TCPWM	4	4	8	8	4	
CAN	なし	なし	2	2	なし	
Bluetooth® LE	なし	なし	なし	なし	4.1/4.2	
USB フルスピードデバイスコントローラ (USB)	なし	なし	なし	有	なし	

(続く)

3 PSOC™ 4 の機能セット

表 5 (続き) PSOC™ 4200 ファミリの特長

特長		PSOC™ 4200	PSOC™ 4200DS	PSOC™ 4200M	PSOC™ 4200L	PSOC™ 4200 BL ¹⁾
クロック	IMO	3 MHz ~ 48 MHz	3 MHz ~ 48 MHz	3 MHz ~ 48 MHz	3 MHz ~ 48 MHz	3 MHz ~ 48 MHz
	ILO	32 kHz	40 kHz	32 kHz	32 kHz	32 kHz
	WCO	なし	なし	32 kHz	32 kHz	32 kHz
	外部水晶発振器 (ECO)	なし	なし	4 MHz ~ 33 MHz	なし	なし
電源監視		POR, BOD, LVD	POR, BOD	POR, BOD, LVD	POR, BOD, LVD	POR, BOD, LVD
対応するキット		CY8CKIT-042 pioneer kit	CY8CKIT-146 prototyping kit	CY8CKIT-044 pioneer kit	CY8CKIT-046 pioneer kit	CY8CKIT-042 Bluetooth® LE pioneer kit
対応する IDE		PSOC™ Creator	PSOC™ Creator	PSOC™ Creator	PSOC™ Creator	PSOC™ Creator

1) PSOC™ 4 Bluetooth® LE ファミリアデバイスの使用を開始するためには、AN91267 を参照してください。

2) IDAC は、CAPSENSE™ を使用しない場合にのみ利用可能です。詳細は、各 PSOC™ 4 アーキテクチャ TRM を参照してください。

表 6 PSOC™ 4500 および PSOC™ 4700 ファミリの特長

特長	PSOC™ 4500S	PSOC™ 4700S
CM0+ CPU	48 MHz Cortex® -M0+	48 MHz Cortex® -M0+
DMA	8 チャンネル	なし
フラッシュメモリ	256 KB	32 KB
SRAM	32 KB	4 KB
GPIO	53	36
CAPSENSE™	1 チャンネル, 52 センサー	1 チャンネル, 35 センサー
MagSense	なし	1 チャンネル
ADC	2 つの 12 ビット, シーケンサ付き 1 MSPS SAR ADC	10 ビット, 16.8 ksps シングルスロープ ADC
オペアンプ (プログラム可能)	4	なし
コンパレータ (ウェイクアップ機能付き低電力)	2	2
IDAC ¹⁾	2 つの 7 ビット	2 つの 7 ビット
スマート I/O ポート	2	2
電源電圧範囲	1.71 V ~ 5.5 V	1.71 V ~ 5.5 V
低消費電力モード	ディープスリープ	1.3 µA
	ハイバネート	150 nA
	ストップ	20 nA
セグメント LCD 駆動	4 COM	8 COM

(続く)

3 PSOC™ 4 の機能セット

表 6 (続き) PSOC™ 4500 および PSOC™ 4700 ファミリの特長

特長		PSOC™ 4500S	PSOC™ 4700S
プログラム可能な I2C, SPI, または UART を備えた SCB		5	2
TCPWM		8	5
モーター制御アクセラレーション (MCA)		2	なし
クロック	IMO	24 MHz ~ 48 MHz	24 MHz ~ 48 MHz
	ILO	40 kHz	40 kHz
	WCO	32 kHz	32 kHz
	ECO	4 MHz ~ 33 MHz	なし
電源監視		POR, BOD	POR, BOD
対応するキット		CY8CKIT-045S pioneer kit	CY8CKIT-148 evaluation kit
対応する IDE		PSOC™ Creator	PSOC™ Creator

1) IDAC は、CAPSENSE™を使用しない場合にのみ利用可能です。詳細は、各 PSOC™ 4 アーキテクチャ TRM を参照してください。

4 PSOC™が MCU より優れている点

4 PSOC™が MCU より優れている点

図 9 は標準的な MCU を示し、CPU (8051 または Arm® Cortex® など) および ADC, DAC, UART, SPI 等の周辺機能および汎用 I/O を含みます。これらはすべて CPU のレジスタ インターフェースにリンクします。MCU 内で、CPU はデバイスの「心臓」です。CPU はセットアップ、データ移動、タイミングなどをすべての処理を管理します。CPU なしでは、MCU は機能しません。

図 10 に示すように、PSOC™はまったく違います。PSOC™では、CPU, アナログ, デジタル, および I/O は、プログラマブルなシステムで同等に重要なリソースです。PSOC™の心臓はシステムの相互接続とプログラマビリティであり、CPU ではありません。周辺機能アナログとデジタルは高度に設定可能な信号マトリックスおよびデータバスメッシュを介して相互接続します。これにより、ユーザーがアプリケーション要件を満たすカスタム設計を作成可能にします。PSOC™をプログラムして MCU をエミュレートできますが、MCU をプログラムして PSOC™をエミュレートできません。

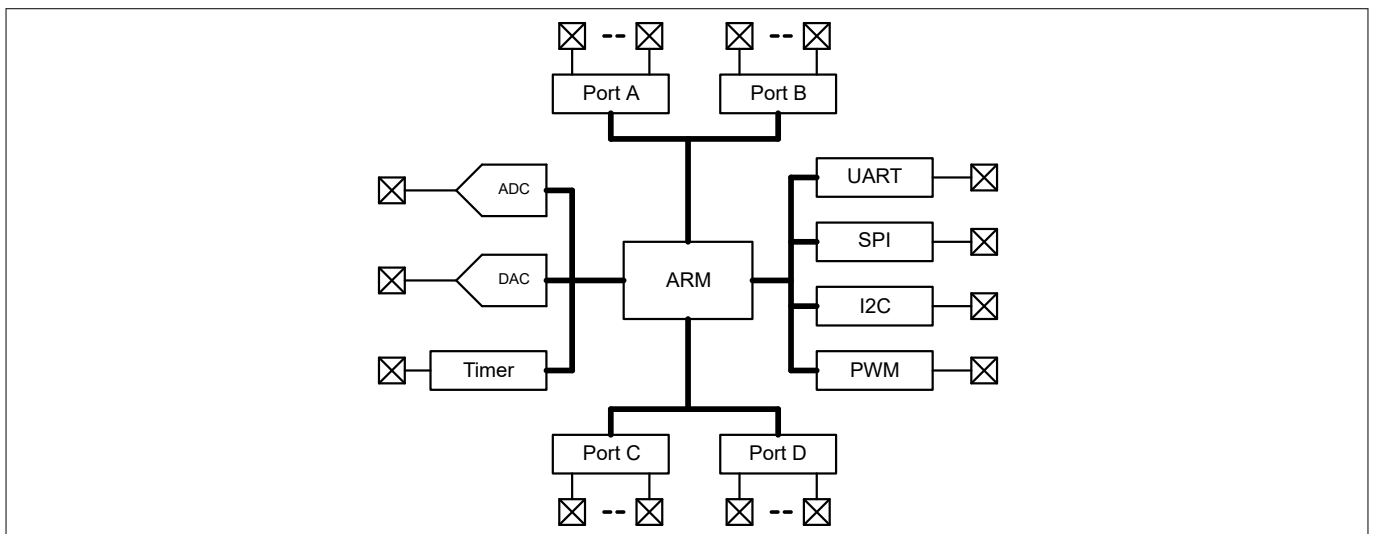


図 9 標準 MCU ブロックダイアグラム

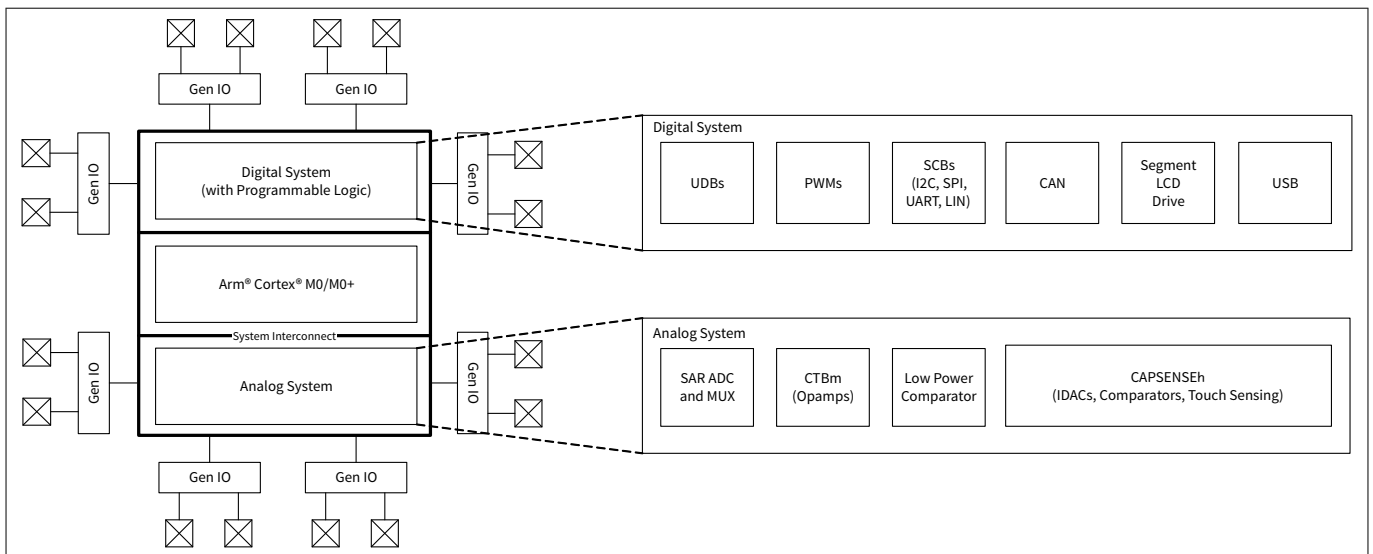


図 10 PSOC™のブロックダイアグラム

標準的な MCU は、ステートマシンを処理し、タイマーをタイミングに使用し、出力ピンを駆動するために、CPU ファームウェアを必要とします。したがって、機能的なパスはほとんどの場合が CPU を介します。しかし、PSOC™では非同期の平行処理が可能です。PSOC™を設定し、CPU から独立して動作する要素を取得できます。このアプリケーションノートに組み込まれるプロジェクトはこの概念を示します。パート 1: 設計の作成で説明するように PSOC™は CPU にコードを書かずに LED を点滅させるように設定されます。

4 PSOC™が MCU より優れている点

4.1 PSOC™ Creator コンポーネントのコンセプト

PSOC™のその他の重要な点は PSOC™ Creator IDE の使用可能性です。PSOC™ Creator では、異なる PSOC™リソースは設計を迅速にビルドするために、回路図にドラッグ アンドドロップできるコンポーネントと呼ばれているグラフィック要素として構成されます。すべての PSOC™のペリフェラルは、検証済み PSOC™ Creator コンポーネント (PWM コンポーネント、ADC コンポーネント、DAC コンポーネント、CAPSENSE™コンポーネント、UART コンポーネントなど) として使用できます。PSOC™ Creator の検証済みコンポーネントの使用可能性は開発時間を著しく短縮します。それにより、グラフィック オプションを用いて、設計も迅速に変更できます。

例えば、標準的なマイクロコントローラーで PWM を設定し、LED を点滅させることは以下のことを含みます。

1. PWM ブロックに対応したレジスタの位置付け
2. 要求される PWM 周期およびデューティ比に基づいて、PWM レジスタに書き込む値を計算
3. PWM レジスタを設定するために数多くのコードラインを書き、ピン駆動モードを設定し、PWM 出力をピンに接続。数多くの MCU は内部ブロックに接続するために、代替のピンは提供しない

PSOC™での同じ機能の実装は、本アプリケーション ノートの後半で説明する簡単な操作です。これから、タイマーに同じ PWM ブロックを再設定する場合は、PSOC™ Creator ではわずかなマウス クリックだけで、それ以上に手間がかかりません。

PSOC™には、ユニバーサル デジタル ブロック (UDB) として知られるプログラマブル デジタル ブロックもあります。PSOC™ Creator はまた、UART, SPI, I2C, タイマー, PWM, カウンター, デジタル ゲート (AND, OR, NOT, XOR) など、UDB からなる様々なコンポーネントを提供します。PSOC™ Creator で UDB を使用し、自分のカスタム ステート マシンおよびデジタル ロジックを作り上げられます。カスタム PSOC™ Creator コンポーネントの作成方法は [PSOC™ Creator Component author guide](#) を参照してください。

5 はじめての PSOC™ 4 設計

5 はじめての PSOC™ 4 設計

ここでは、

- 単純な PSOC™ 4 アプリケーションを構築し、それを開発キットにプログラムする方法を示します。
- PSOC™ 4 デザインテクニックや ModusToolbox™ ソフトウェアをさまざまな IDE で使用方法を簡単に学びます。

注: サポートされている IDE ならどれでも使えますが、このセクションでは Eclipse IDE を例にしています。

5.1 前提条件

開始する前に、PSOC™ 4 製品ラインに適切な開発キットがあり、必要なソフトウェアがインストールされていることを確認してください。プロジェクトの作成中に、GitHub リポジトリへのインターネットアクセスも必要です。

5.1.1 ハードウェア

この設計をテストするためには、プログラマが統合された、表 7 にリストされているキットの 1 つが必要です。

表 7 PSOC™ 4 パイオニアキット, プロトタイピングキット, およびサポートされているデバイスのリスト

キット名	キット形式	対応するデバイスファミリ	製品番号
CY8CKIT-145	プロトタイピングキット	PSOC™ 4000S	CY8C4045AZI-S413
CY8CKIT-149	プロトタイピングキット	PSOC™ 4100S Plus	CY8C4147AZI-S475
CY8CKIT-041S-Max	パイオニアキット	PSOC™ 4100S Max	CY8C4149AZI-S598
CY8CKIT-040T	評価キット	PSOC™ 4000T	CY8C4046LQI-T452
CY8CPROTO-040T	プロトタイピングキット	PSOC™ 4000T	CY8C4046LQI-T452
CY8CPROTO-041TP	プロトタイピングキット	PSOC™ 4100T Plus	CY8C4147AZQ-T495

5.1.2 ソフトウェア

ModusToolbox™ 3.2 以降

ソフトウェアをインストールした後、[ModusToolbox™ tools package user guide](#) を参照し、本ソフトウェアの概要を確認してください。

5.2 これらの手順の使用

これらの手順は、いくつかのセクションにグループ化されています。各セクションでは、アプリケーション開発ワークフローのフェーズについて説明します。主なセクションは次のとおりです。

1. [ボードの BSP の作成](#)
2. [新しいアプリケーションの作成](#)
3. [設計の参照と変更](#)
4. [ファームウェアの書き込み](#)

5 はじめての PSOC™ 4 設計

5. アプリケーションのビルド
6. デバイスのプログラム
7. 設計のテスト

ModusToolbox™ を使用したプロジェクトの開発に慣れている場合は、「Hello World」や「Blinky」など、キットでサポートされているスターター アプリケーションの 1 つを直接使用できます。このスターター アプリケーションは完全なプロジェクトであり、追加のコードや設定を必要とせず、サポートされているキット上で実行できます。この設計は、表 7 にリストされているキット用に開発されています。アプリケーションの作成中に適切なキットを選択することにより、このサンプルをテストできます。README.md ファイルの手順を確認し、サンプルコードでその手順がどのように実装されているかを確認してください。

このアプリケーション ノートの手順に従って別のアプリケーションを最初から作成する場合でも、このサンプルコードを参考として使用できます。

5.3 設計について

この設計では、PSOC™ 4 MCU を使用して LED を点滅させます。UART をサポートするキットでは、「Hello World」メッセージがシリアル ポート ストリームに出力されます。

5.4 ボードの BSP の作成

すべての ModusToolbox™ アプリケーションにはターゲット BSP が必要です。インフィニオンは、当社のすべてのキット (表 7 に記載されているキットを含む) および各チップ アーキテクチャ用の BSP を提供しており、出発点として使用できます。独自のハードウェアを使用する場合、ハードウェアに合わせてインフィニオンの BSP を変更できます。BSP アシスタントは、プロセスを簡素化するのに役立ちます。独自のハードウェア用に開発する場合は、ModusToolbox™ BSP Assistant user guide に記載されている手順に従ってカスタム BSP を作成してください。Device configurator を開くセクションは、カスタム BSP のコンフィギュレーションを編集する方法について説明します。

5.5 新しいアプリケーションの作成

ここでは新しいアプリケーションのプロセスの概要について順に説明します。Empty PSOC4 App スターター アプリケーションを使用し、Hello World スターター アプリケーションの機能を手動で追加します。説明では ModusToolbox™ 用の Eclipse IDE を使用していますが、どの IDE やコマンドラインでも使用できます。

ModusToolbox™ を使用したプロジェクト開発に精通している場合は、「Hello World」スターター アプリケーションを直接使用できます。すべてのファームウェアが対応しているキット用に書かれた完全な設計です。手順を確認し、サンプルコードでその手順がどのように実装されているかを確認してください。

ゼロから始め、本アプリケーション ノートに記載されているすべての説明に従う場合は、Hello World のサンプルコードを参考にしてください。

Dashboard 3.2 アプリケーションを起動して開始してください。

注: **Dashboard 3.2** アプリケーションは、スターターアプリケーションをマシンに正常に複製するために、インターネットにアクセスする必要があることに注意してください。

Dashboard 3.2 アプリケーションは、ドキュメントやトレーニング資料への簡単なアクセス、アプリケーションの作成や BSP の作成/編集のためのシンプルなパスにより、様々なツールの使用を開始するのに役立ちます。

1. Dashboard 3.2 アプリケーションを開いてください。

Dashboard 3.2 アプリケーションを開くには、以下のいずれかの操作を行ってください。

- **Windows:** [ModusToolbox installation path]/tools_3.2/dashboard/dashboard.exe に移動するか、Windows のスタートメニューから「ダッシュボード」を選択してください。
- **Linux:** [ModusToolbox installation path]/tools_3.2/dashboard で実行ファイルを実行してください。
- **macOS:** ダッシュボードアプリを実行してください。

5 はじめての PSOC™ 4 設計

2. **Dashboard 3.2** ウィンドウの右側で、**Target IDE** ドロップダウンリストから **Eclipse IDE for ModusToolbox™** を選択し、**Launch Eclipse IDE for ModusToolbox™** をクリックしてください。

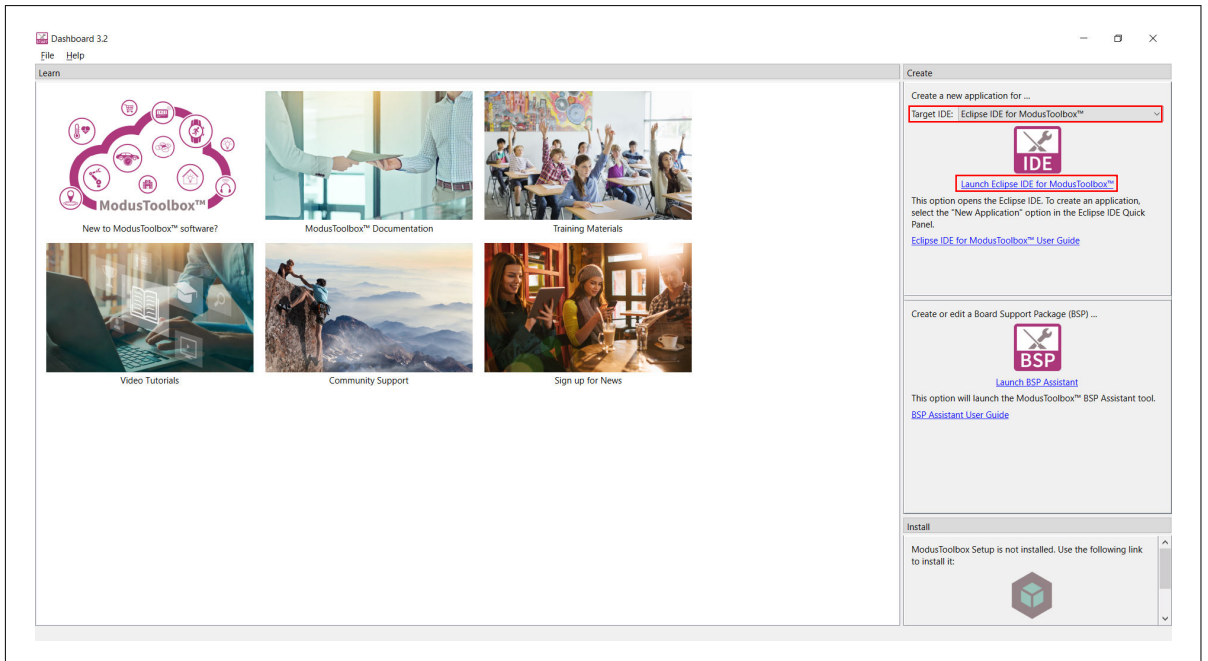


図 11 ダッシュボード 3.2 アプリケーション

3. **新しいワークスペースの選択。**

ModusToolbox™ の Eclipse IDE の起動時、ワークスペース ディレクトリとして使用するディレクトリを選択するダイアログが表示されます。ワークスペース ディレクトリは、ワークスペースの設定と開発結果の資産を格納するために使用されます。以下の図に示す、ように、**Browse** ボタンをクリックして、既存の空のディレクトリを選択できます。または、ワークスペース ディレクトリとして使用するディレクトリ名および完全パスを入力すると、IDE がそのディレクトリを作成します。

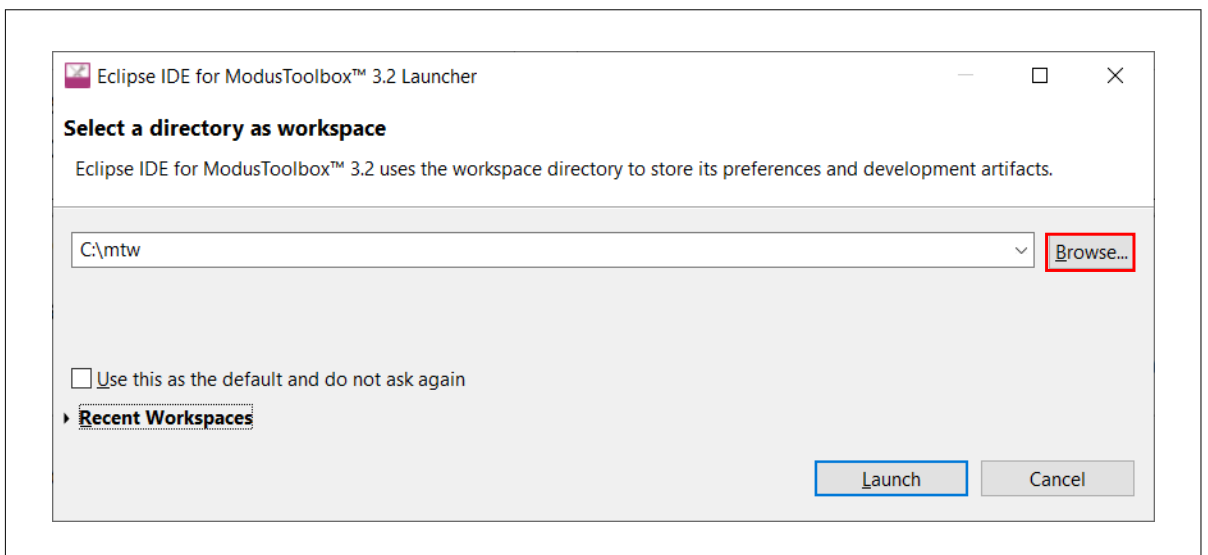


図 12 ワークスペースとしてのディレクトリの選択

4. **ModusToolbox™ IDE で新しいアプリケーションの作成。**

- a. Quick Panel の Start グループで **New Application** をクリックします。
- b. または、下図のように **File > New > ModusToolbox™ Application** を選択することもできます。。
Project Creator が開きます。

5 はじめての PSOC™ 4 設計

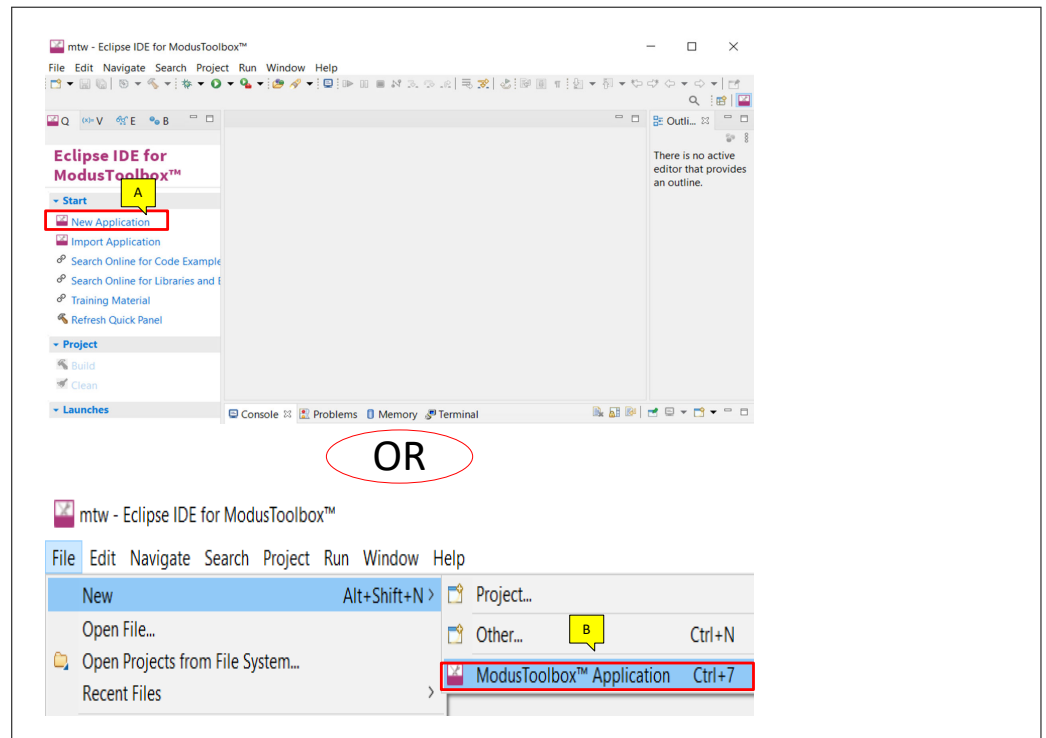


図 13 ModusToolbox™ IDE で新しいアプリケーションの作成

5. ターゲットの PSOC™ 4 開発キットを選択してください。

ModusToolbox™ は、新しいアプリケーションダイアログで指定された開発キットのさまざまなワークスペース/プロジェクトオプションを設定する BSP を提供することにより、開発プロセスをスピードアップします。

- a. **Choose Board Support Package (BSP)** ダイアログで、使用する **Kit Name** を選択してください。以下の手順では **CY8CPROTO-040T** の使用を想定しています。この手順については [図 14](#) を参照してください。
- b. **Next** をクリックしてください。

5 はじめての PSOC™ 4 設計

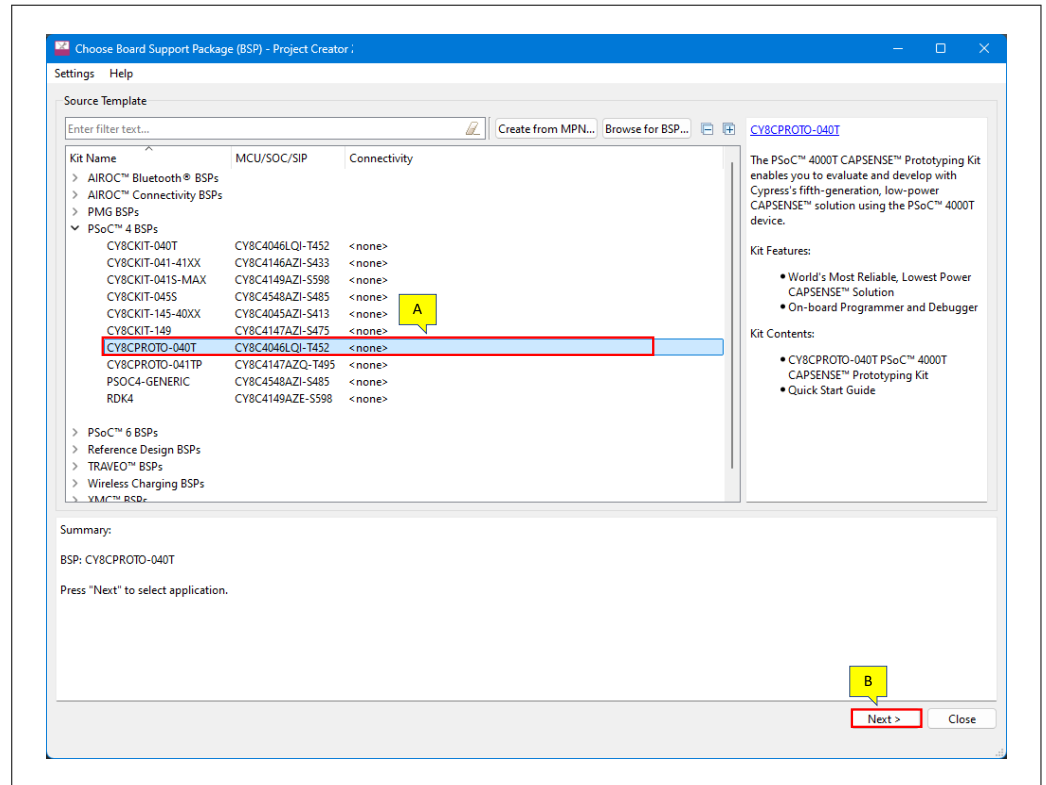


図 14 ターゲットハードウェアの選択

- c. 下図のように、**Select Application** ダイアログで、**Empty App** starter アプリケーションを選択してください。
- d. **Name** フィールドには、**Hello_World** などのアプリケーションの名前を入力してください。必要に応じて、デフォルト名のままにすることもできます。
- e. に示すように **Create** をクリックしてアプリケーションを作成してください。プロジェクトが正常に作成されたら、Project Creator が自動的に閉じるのを待ってください。

5 はじめての PSOC™ 4 設計

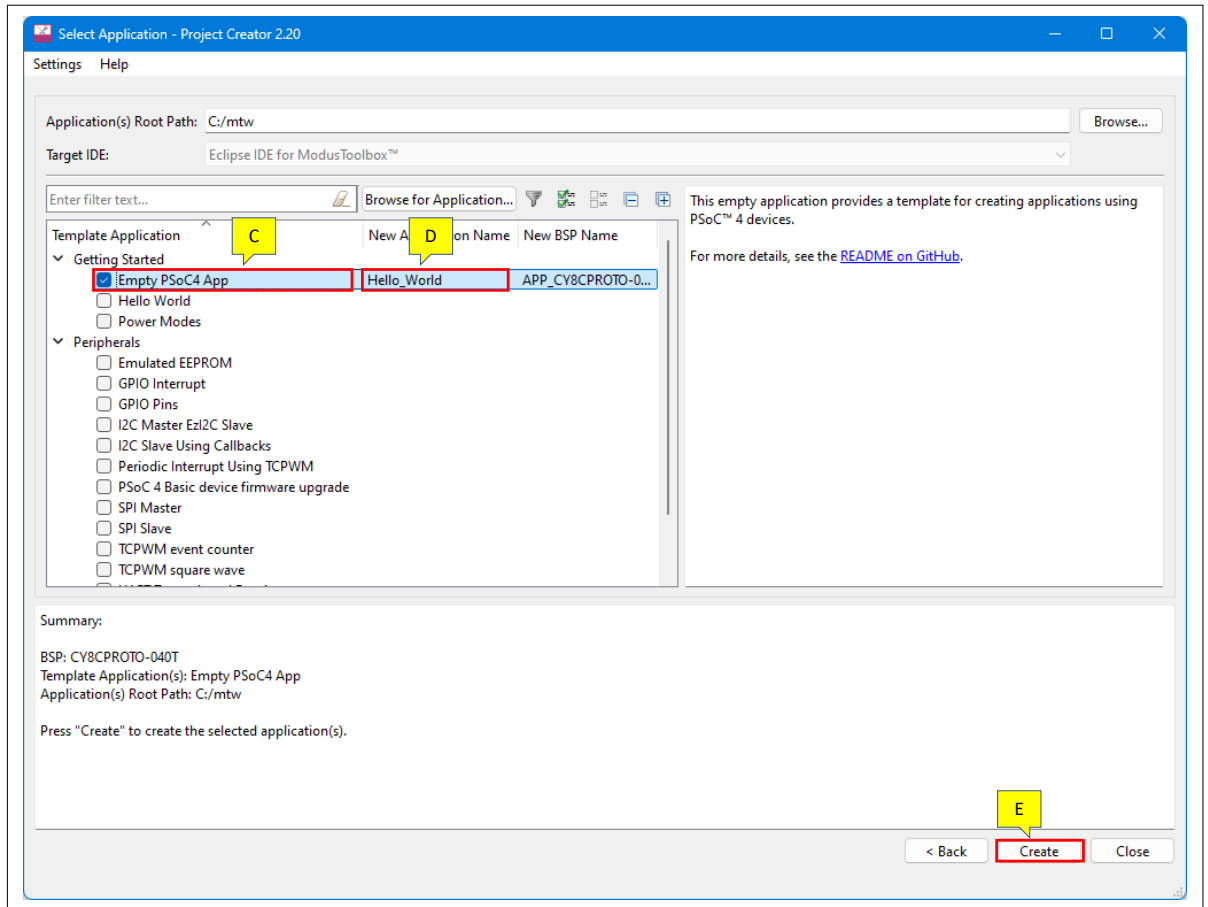


図 15 スターターアプリケーションの選択

これで、PSOC™ 4 用の新しい ModusToolbox™アプリケーションが正常に作成されました。

BSP は、CYW43439KUBG Wi-Fi/Bluetooth®無線とともに、PSOC™ 4 CY8CPROTO-040T Prototyping Kit に搭載されたデフォルト デバイスとして CY8C4046LQI-T452 を使用します。

PSOC™ 4 または異なる PSOC™ 4 部品番号に基づくカスタム ハードウェアを使用する場合は、[ボードの BSP の作成](#)または [BSP Assistant user guide](#) を参照してください。

5.6 設計の参照と変更

図 16 に、アプリケーションプロジェクトの構造を表示する ModusToolbox™プロジェクトエクスプローラーを示します。

5 はじめての PSOC™ 4 設計

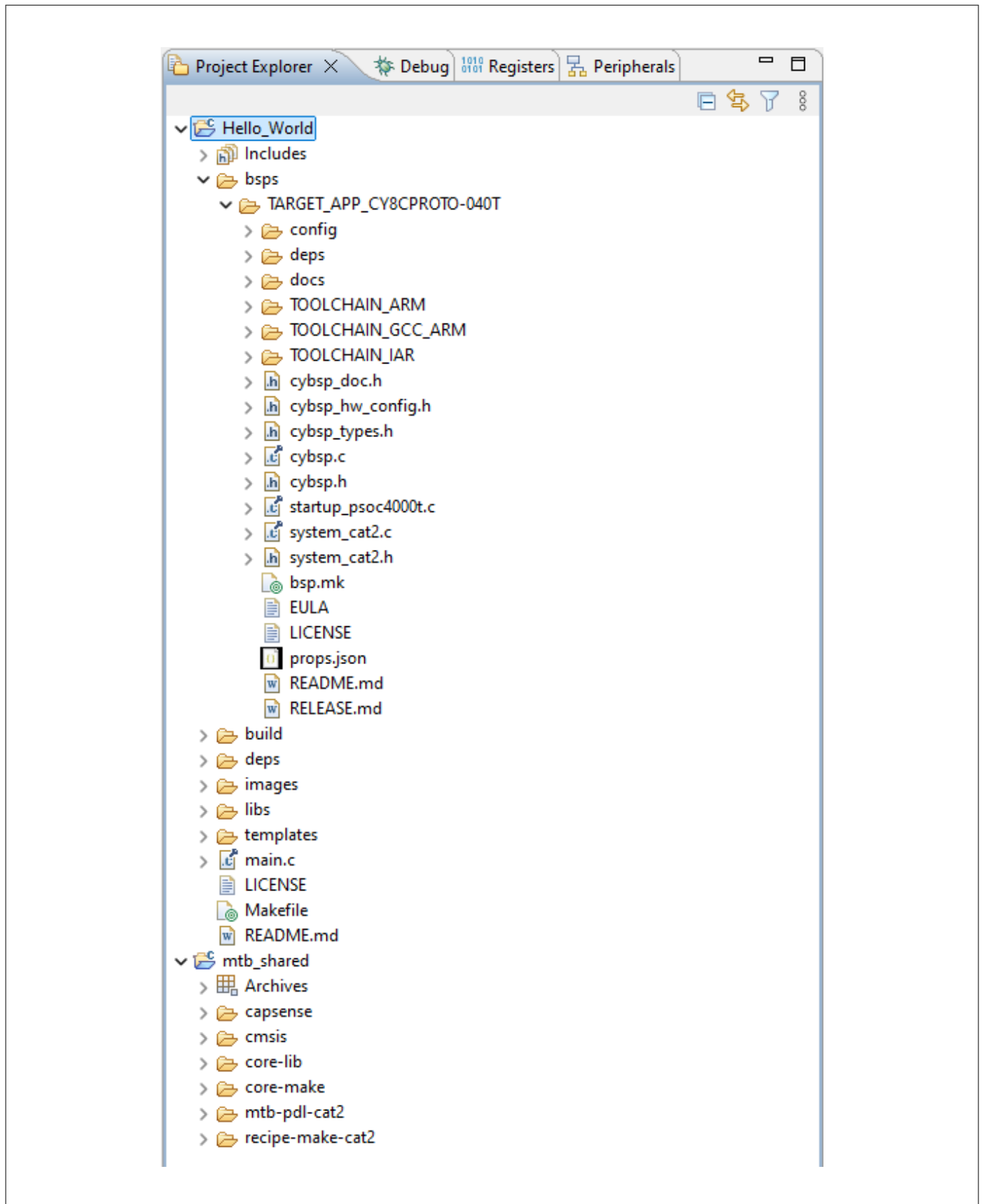


図 16 プロジェクトエクスプローラー外観

ファイルとフォルダの詳細を理解するには、[ModusToolbox™ tools package user guide](#) のセクション 2.4.2 を参照してください。

5 はじめての PSOC™ 4 設計

5.6.1 Device configurator を開く

選択した BSP によってロードされるデバイス設定を変更できます。例えば、デフォルトのキット BSP では UART が有効になっていません。以下の手順は、「Hello World」印刷操作に UART ペリフェラルを有効にする方法と、PSOC™ 4 キットのユーザー LED を有効にする方法を示しています。

1. 図 17 に示すように、**BSP Configurators** セクションの下の **Quick Panel** で **Device configurator** リンクをクリックします。これにより、Device Configurator ダイアログが開きます (図 18)。他のコンフィギュレーターを開いて、必要に応じて設定することもできます。

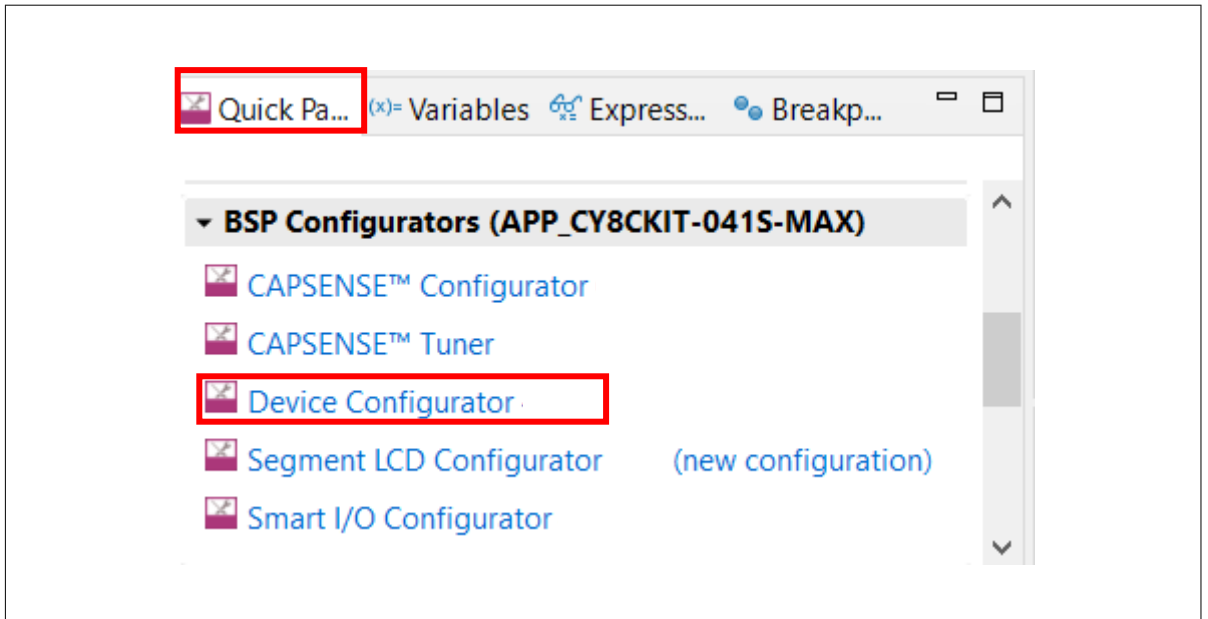


図 17 Device configurator を開く

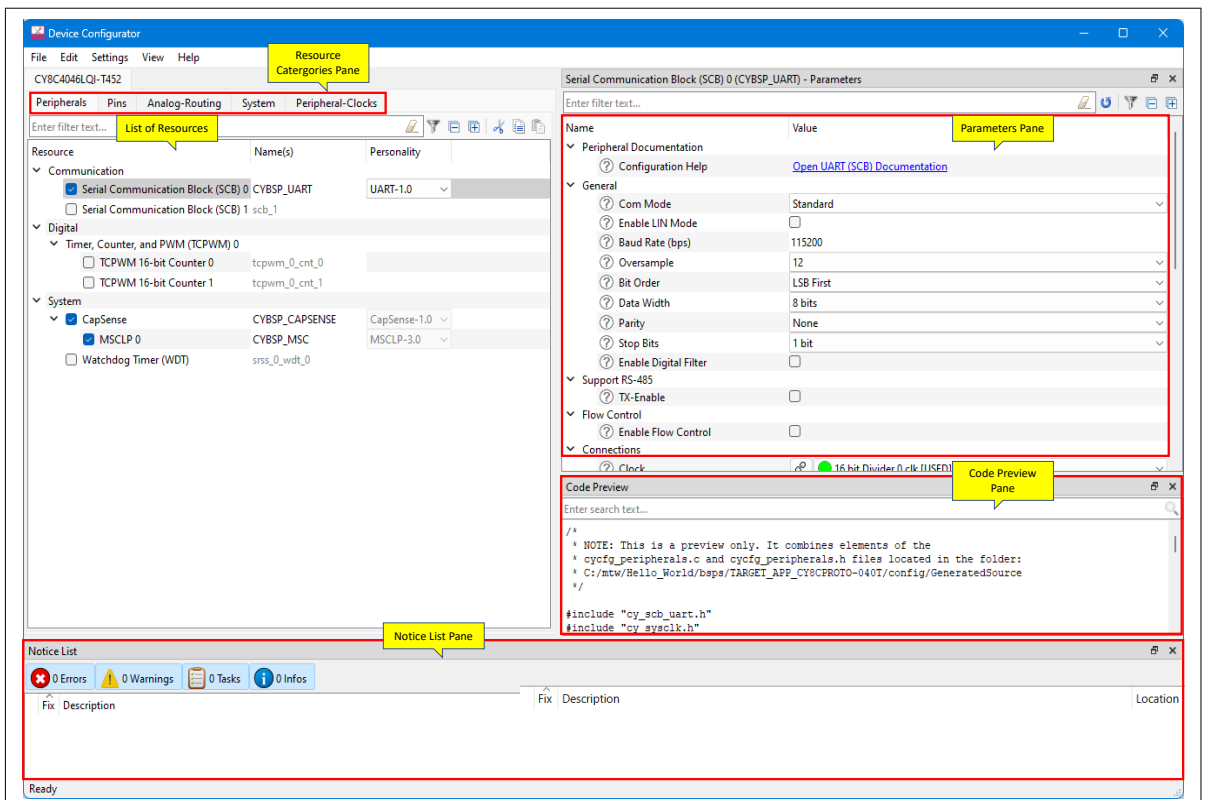


図 18 design.modus の概要

5 はじめての PSOC™ 4 設計

Device Configurator ダイアログの **Resource Categories** ペインから、周辺機器、ピン、クロックなどのさまざまなリソースから選択できます。

Peripherals タブには、デバイスで使用可能な周辺機器のリストが表示されます。**Personality** 列の設定は、リソースの動作を定義します。例えば、**Serial Communication Block (SCB)** リソースには、パーソナリティとして **EZ12C, I2C, SPI**, または **UART** を含めることができます。**Name(s)** フィールドは、ファームウェア開発で使用されるリソースのユーザー定義の名前です。ModusToolbox™ では、さまざまな実行時設定に対応してさまざまなエイリアスをピンに割り当てる機能が提供されているため、1 つ以上のピンの名前を個々に指定できます。リストされた名前は、スペースなしのカンマで区切らなければならないことに注意してください。

Parameters ペインで、有効な各リソースおよび選択したパーソナリティの構成パラメーターを入力できます。**Code Preview** ペインには、選択した設定パラメーターに対して生成された設定コードが表示されます。このコードは、GeneratedSource フォルダの `cycfg_` ファイルに入力されます。設定に起因するエラー、警告、および情報メッセージは、**Notices List** ペインに表示されます。

2. 次の手順で説明するように、UART ペリフェラルを有効にし、対応するクロックを設定します。

注: *CY8CKIT-040T* キットの場合、このキットでは UART がサポートされていないため、これらの手順をスキップしてください。

- a. **Resources Categories** ペインの **Peripherals** タブに移動し、**Serial Communication Block (SCB) 0** を有効にし、パーソナリティを UART-1.0 に、名前を CYBSP_UART に設定します。図 19 では、キットに必要な Rx/Tx ピンに基づいて SCB インスタンス 1 が選択されています。図 20 に示すように、Rx および Tx 設定パラメーターのドロップダウンには、SCB 1 でサポートされるピンが表示されます。各 SCB インスタンスで使用できるピンについては、[デバイス データシート](#) を参照してください。

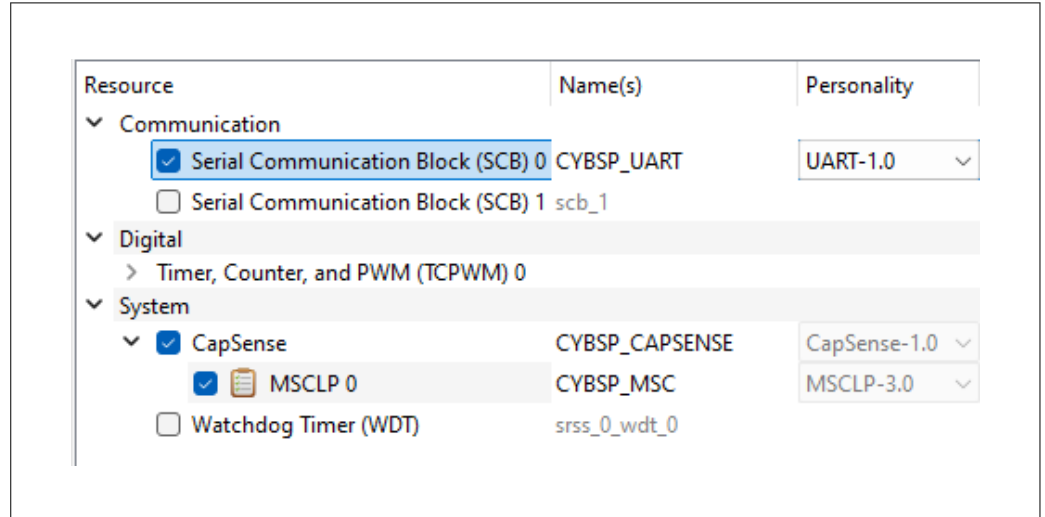


図 19 SCB を UART として有効化

- b. **Parameter** ペインで、**Clock, Rx**, および **Tx** などの必要な設定パラメーターを設定します。他のパラメーターについてはデフォルト値を保持してください。

5 はじめての PSOC™ 4 設計

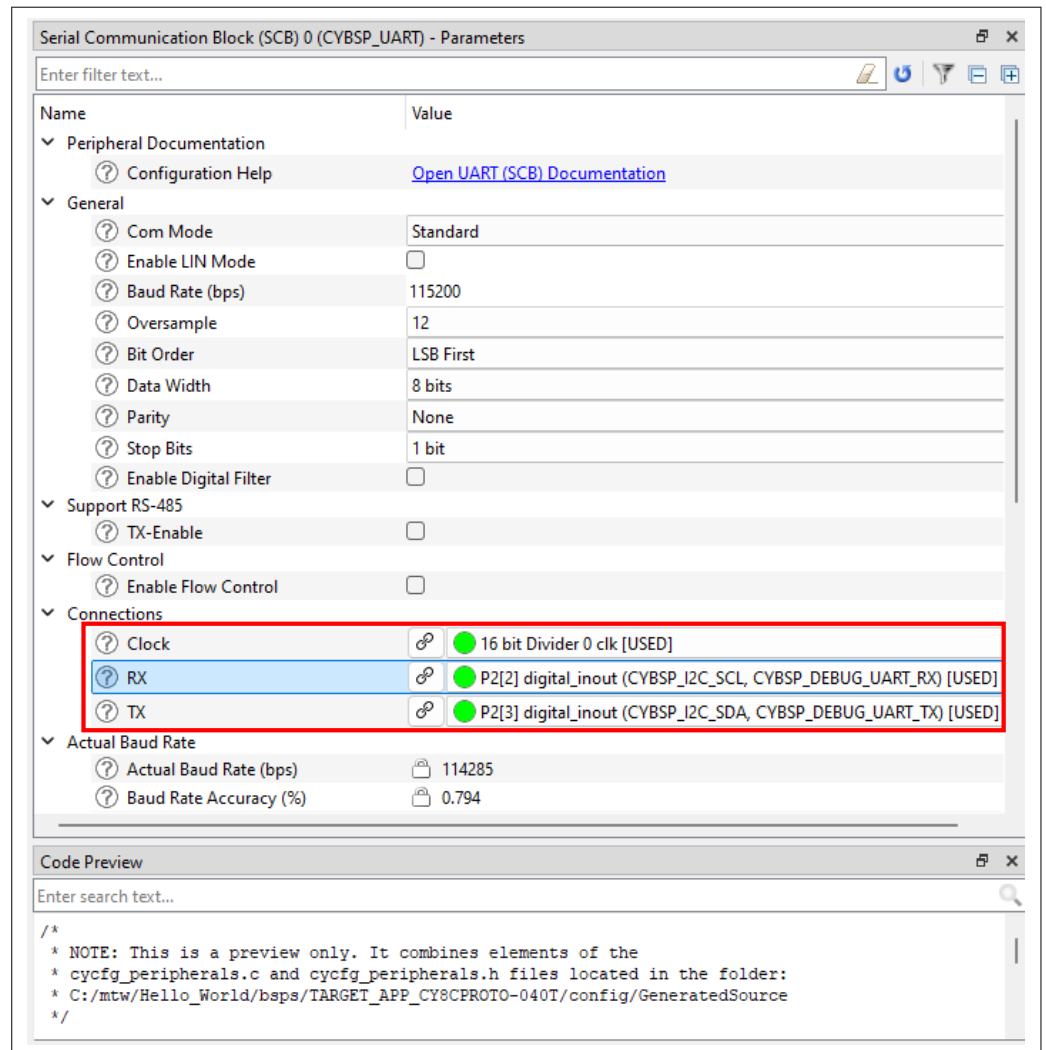


図 20 UART 設定パラメーターの設定

- c. **Resource Categories** ペインの **Peripheral-Clocks** タブに移動して、UART ペリフェラルのクロックを設定してください。

5 はじめての PSOC™ 4 設計

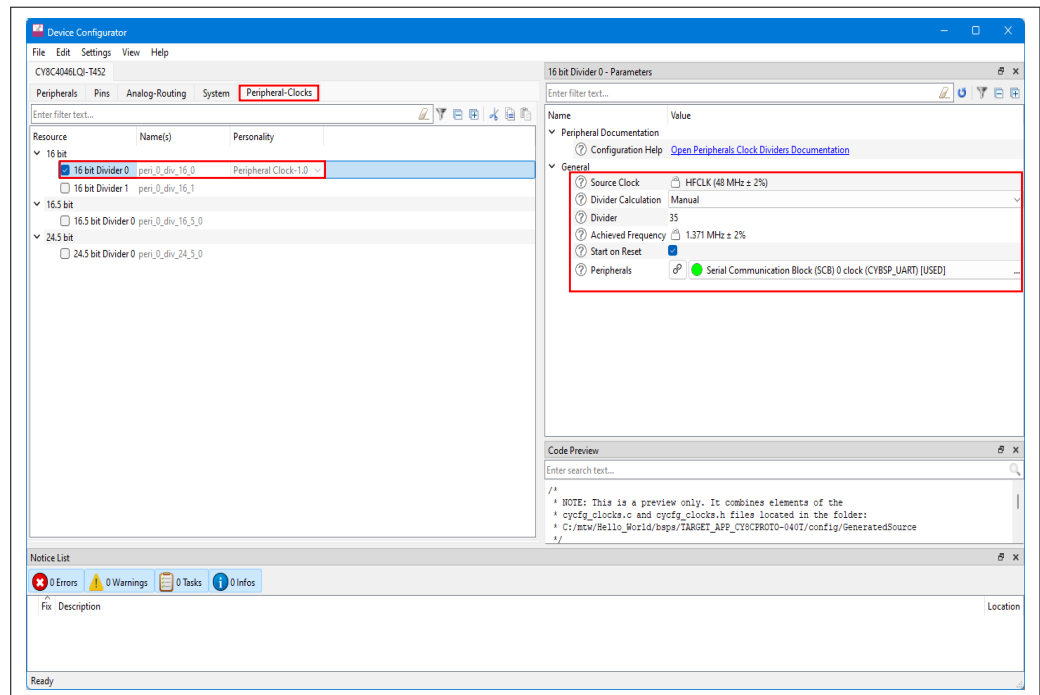


図 21 クロックの設定

ヒント: **Code Preview** ペインに移動して、生成されたコードをプレビューしてください。

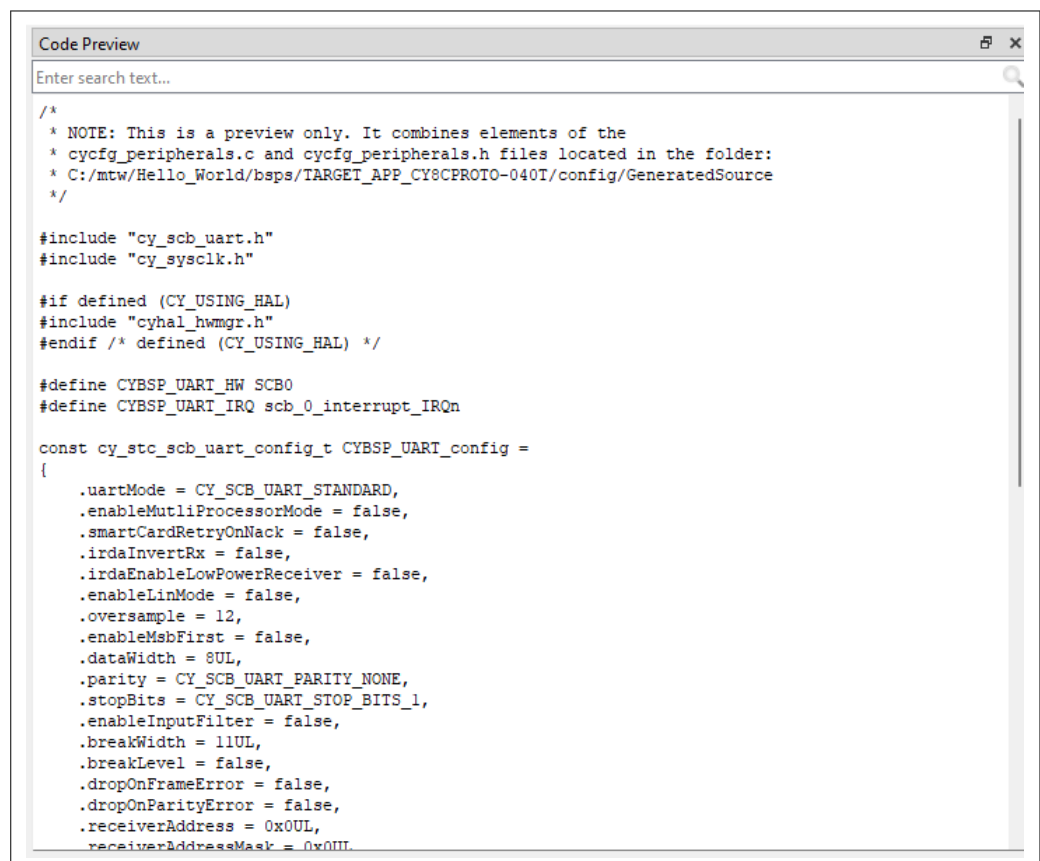


図 22 コードプレビューペイン

5 はじめての PSOC™ 4 設計

3. **Resource Categories** ペインの **Pins** タブに移動して、キットのユーザー LED に接続されている GPIO を設定してください。別の PSOC™ 4 キットを使用している場合は、表 8 を参照してください。図 23 に示すように GPIO を設定してください。

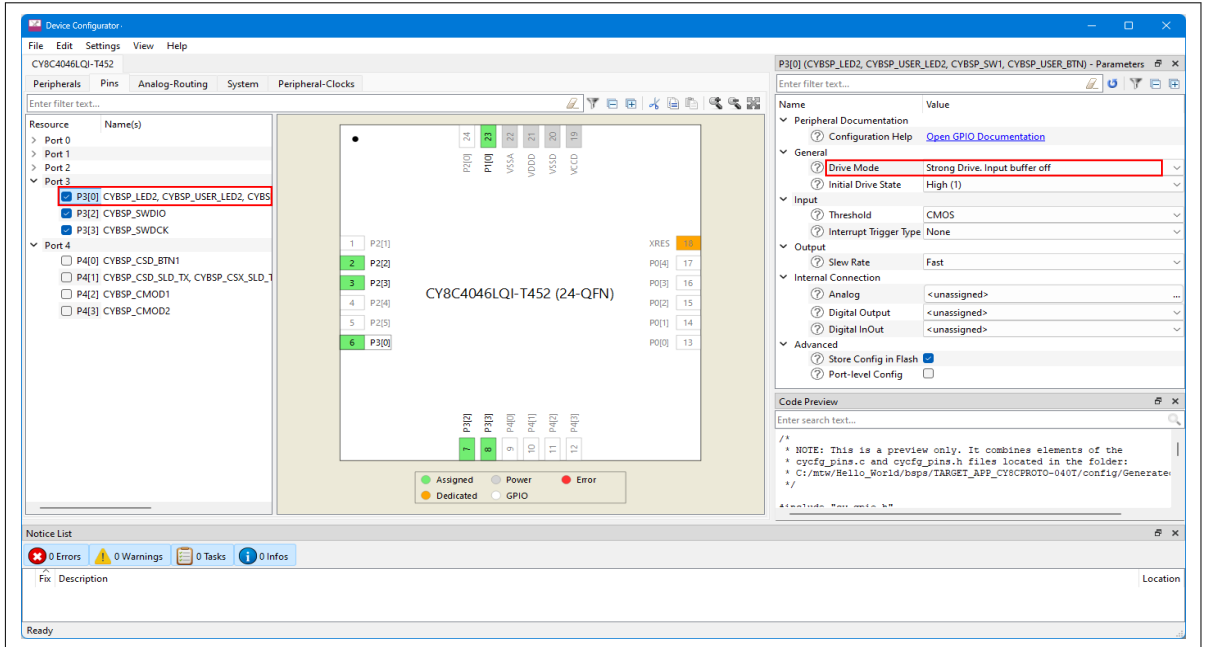


図 23 GPIO の設定

表 8 PSOC™ 4 キット全体のピンマッピング表

機能	CY8CKIT-145(PSOC™ 4000S)	CY8CKIT-149(PSOC™ 4100S Plus)	CY8CKIT-041 S-MAX(PSOC™ 4100S Max)	CY8CKIT-040T PSOC™ 4000T CAPSENSE™ Evaluation Kit	CY8CPROTO-040T PSOC™ 4000T CAPSENSE™ Prototyping Kit	CY8CPROTO-041TP CAPSENSE™ Prototyping Kit
ユーザー LED	P2[5]	P3[4]	P7[3]	P4[0]	1[0], 3[0]	5[4], 5[5]

4. キットのデフォルトで有効になっている可能性がある他のペリフェラルおよびピンを無効にします (つまり、チェックを外します)。
5. 電力, 周波数, およびクロックなどの他のリソースは、Resource Categories ペインの System タブで設定できます。デフォルトの電力設定を図 24 に示します。この設計では、それ以上の変更は必要ありません。

5 はじめての PSOC™ 4 設計

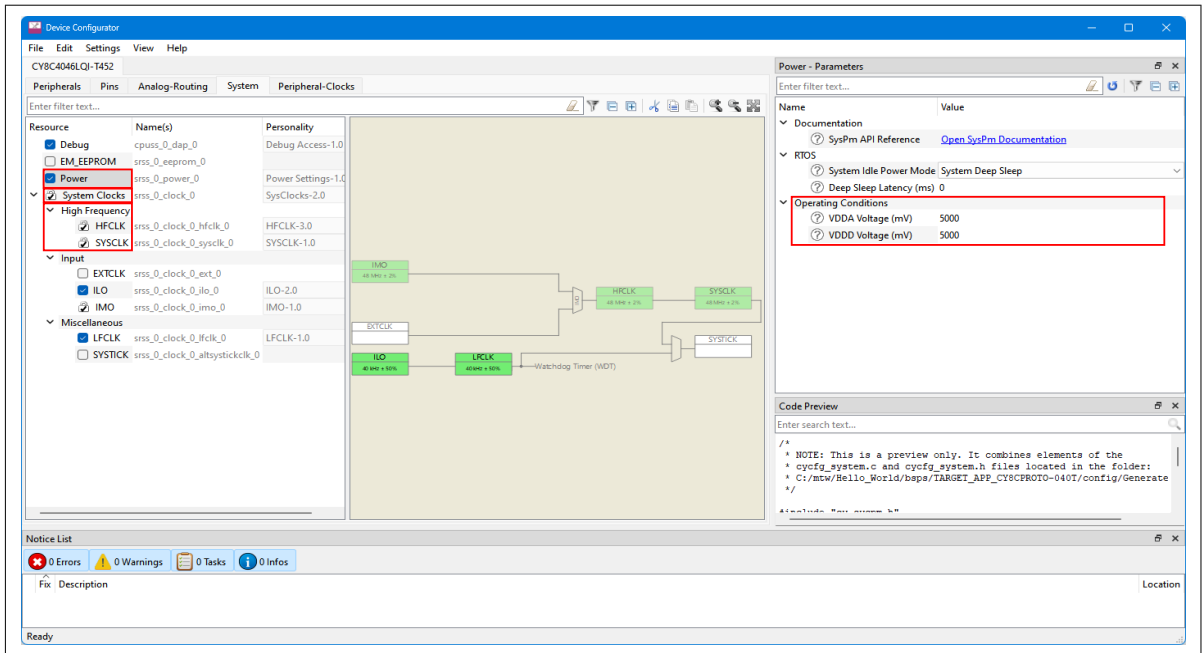


図 24 システム設定

6. **File > Save** に移動するか、**Ctrl]+[S** を押して設定を保存し、GeneratedSource フォルダにソースコードを生成してください。

5.7 ファームウェアの書き込み

この設計は、PSOC™ 4 の CM0+ CPU を使用して、UART 通信と LED 制御の 2 つのタスクを実行します。CM0+ CPU は、UART を使用して「Hello World」メッセージをシリアルポートストリームに出力し、キット上のユーザー LED を点滅させます。

Empty PSOC™ 4 スターターアプリケーションを使用している場合は、ここで提供されているプログラムコードからアプリケーションプロジェクトの main.c ファイルにそれぞれのソースコードをコピーできます。**Hello World** サンプルコードを使用している場合、必要なファイルはすでにアプリケーション内にあります。

ファームウェアフロー

ここでは、アプリケーションの main.c ファイルのコードについて説明します。

この例では、CM0+ CPU がリセットから抜け出し、リソースの初期化を実行します。システムクロック、ピン、クロックから周辺機器への接続、およびその他のプラットフォームリソースを設定します。

クロックとシステムリソースは、BSP 初期化関数によって初期化されます。PDL 関数は、UART 周辺機器を設定して有効にするために使用されます。UART は、ターミナルエミュレータに「Hello World」メッセージを出力します。オンボードの KitProg3 は、仮想 COM ポートを作成するための USB-UART ブリッジとして機能します。ソフトウェア遅延を伴う無限 FOR ループは、ユーザー LED を定期的に切り替えるために使用されます。

アプリケーションコードは、BSP/PDL 関数を使用して目的の機能を実行することに注意してください。

- cybsp_init() - この BSP 関数は、システムクロックと電力レギュレーターを含みますが、これらに限定されないデバイスのシステムリソースを初期化します。
- Cy_SCB_UART_Init() - この PDL 関数は、UART 操作に SCB ブロックを初期化します。設定構造 CYBSP_UART_config は、UART を設定するためのこの関数のパラメーターとして使用されます。この構造は、適用された設定に基づいて design.modus によって自動生成されます。
- Cy_SCB_UART_Enable() - この PDL 関数は、UART 操作の SCB ブロックを有効にします。
- Cy_SCB_UART_PutString() - この PDL 関数は、NULL で終了する文字列を UART TX FIFO に配置します。
- Cy_GPIO_Inv() - この PDL 関数は、ピン出力ロジック状態を現在の出力ロジック状態の逆に設定します。
- Cy_SysLib_Delay() - この PDL 関数は、指定されたミリ秒数だけ遅延します。

5 はじめての PSOC™ 4 設計

該当する場合、[Code Listing 1](#) および [Code Listing 2](#) をアプリケーションプロジェクトの main.c にコピーしてください。

5 はじめての PSOC™ 4 設計

Code Listing 1 UART 通信を使用したコード スニペット (main.c) (CY8CKIT-040T 以外のすべてのキットに使用)

```

/*****
 * Include header files
 *****/
#include "cy_pdl.h"
#include "cybsp.h"

/*****
 * Macros
 *****/
#define LED_DELAY_MS          (500u)
#define CY_ASSERT_FAILED     (0u)
/*****
 * Function Name: main
 *****/
* Summary:
 * System entrance point. This function performs
 * - initial setup of device
 * - configure the SCB block as UART interface
 * - prints out "Hello World" via UART interface
 * - Blinks an LED under firmware control at 1 Hz
 *
 * Parameters:
 * none
 *
 * Return:
 * int
 *
 *****/
int main(void)
{
    cy_rslt_t result;

    cy_stc_scb_uart_context_t CYBSP_UART_context;

    /* Initialize the device and board peripherals */
    result = cybsp_init();

    /* Board init failed. Stop program execution */
    if (result != CY_RSLT_SUCCESS)
    {
        CY_ASSERT(CY_ASSERT_FAILED);
    }

    /* Configure and enable the UART peripheral */
    Cy_SCB_UART_Init(CYBSP_UART_HW, &CYBSP_UART_config, &CYBSP_UART_context);
    Cy_SCB_UART_Enable(CYBSP_UART_HW);

    /* Enable global interrupts */
    __enable_irq();

```

5 はじめての PSOC™ 4 設計

```

/* Send a string over serial terminal */
Cy_SCB_UART_PutString(CYBSP_UART_HW, "Hello world\r\n");

for(;;)
{
    /* Toggle the user LED state */
    Cy_GPIO_Inv(CYBSP_USER_LED1_PORT, CYBSP_USER_LED1_PIN);

    /* Wait for 0.5 seconds */
    Cy_SysLib_Delay(LED_DELAY_MS);
}
    
```

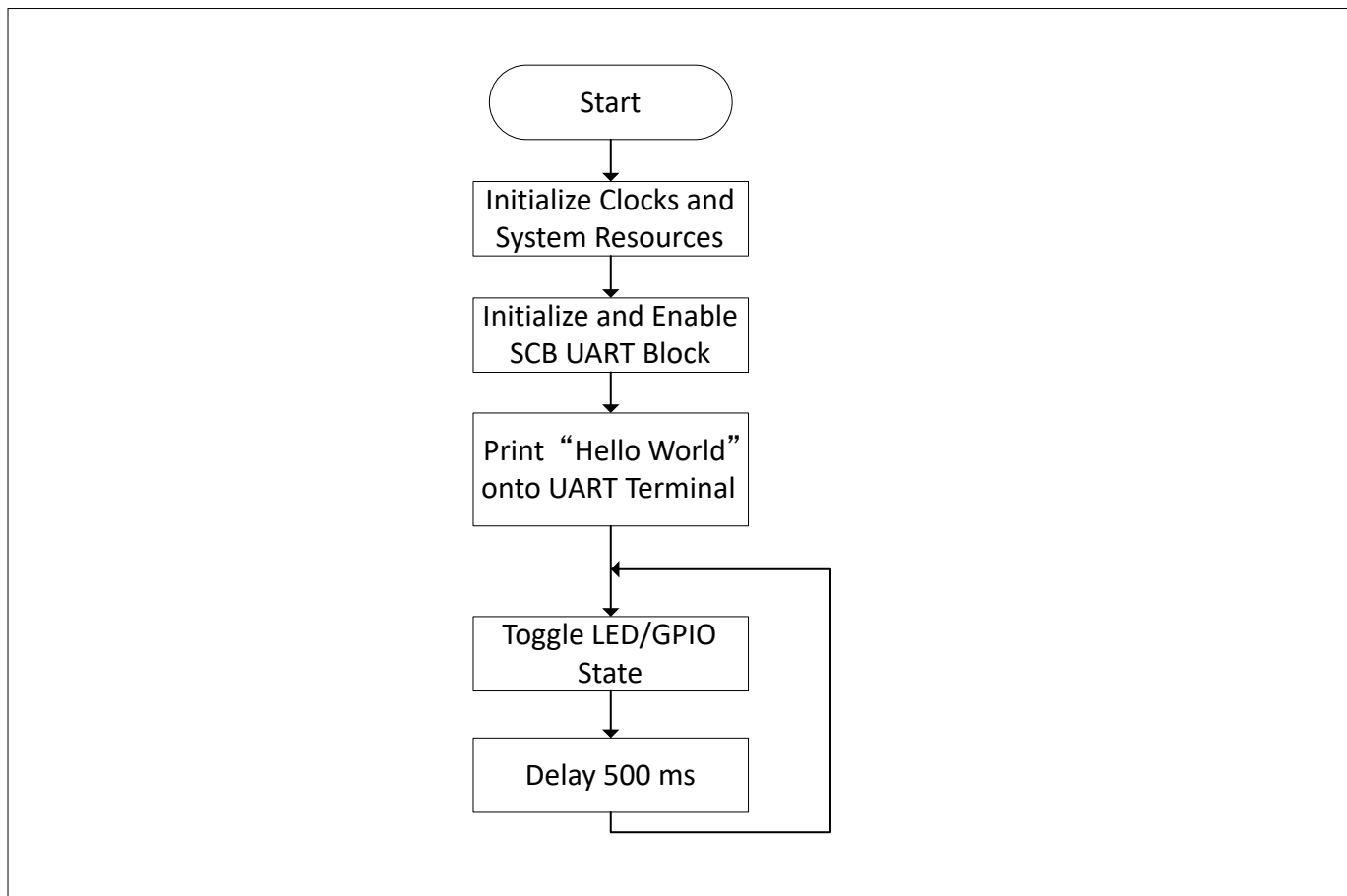


図 25 ファームウェアフローチャート

5 はじめての PSOC™ 4 設計

Code Listing 2 UART 通信を使用しないコード スニペット (main.c) (CY8CKIT-040T を含むあらゆるキットに使用)

```
/******  
 * Include header files  
*****/  
#include "cy_pd1.h"  
#include "cybsp.h"  
  
/******  
 * Macros  
*****/  
#define LED_DELAY_MS          (500u)  
#define CY_ASSERT_FAILED     (0u)  
  
/******  
 * Function Name: main  
*****  
 * Summary:  
 * System entrance point. This function performs  
 * - initial setup of device  
 * - Blinks an LED under firmware control at 1 Hz  
 *  
 * Parameters:  
 * none  
 *  
 * Return:  
 * int  
 *  
*****/  
int main(void)  
{  
    cy_rslt_t result;  
  
    /* Initialize the device and board peripherals */  
    result = cybsp_init();  
  
    /* Board init failed. Stop program execution */  
    if (result != CY_RSLT_SUCCESS)  
    {  
        CY_ASSERT(CY_ASSERT_FAILED);  
    }  
  
    /* Enable global interrupts */  
    __enable_irq();  
  
    for(;;)  
    {  
        /* Toggle the user LED state */  
        Cy_GPIO_Inv(CYBSP_USER_LED1_PORT, CYBSP_USER_LED1_PIN);  
  
        /* Wait for 0.5 seconds */  
        Cy_SysLib_Delay(LED_DELAY_MS);  
    }  
}
```

5 はじめての PSOC™ 4 設計

```

}
}
    
```

これは、サンプルコードでファームウェアがどのように機能するかをまとめたものです。詳細は、ソースファイルを調べてください。

5.8 アプリケーションのビルド

ここでは、アプリケーションのビルド方法を説明します。

1. Project Explorer ウィンドウでアプリケーションプロジェクトを選択し、Quick Panel の<name>グループの下にある **Build <name> Application** ショートカットをクリックしてください。これによって選択されたビルド設定を使用して、アプリケーションを構成するすべてのプロジェクトをコンパイル/リンクします。
2. **Console** ビューには、ビルド操作の結果が一覧表示されます。

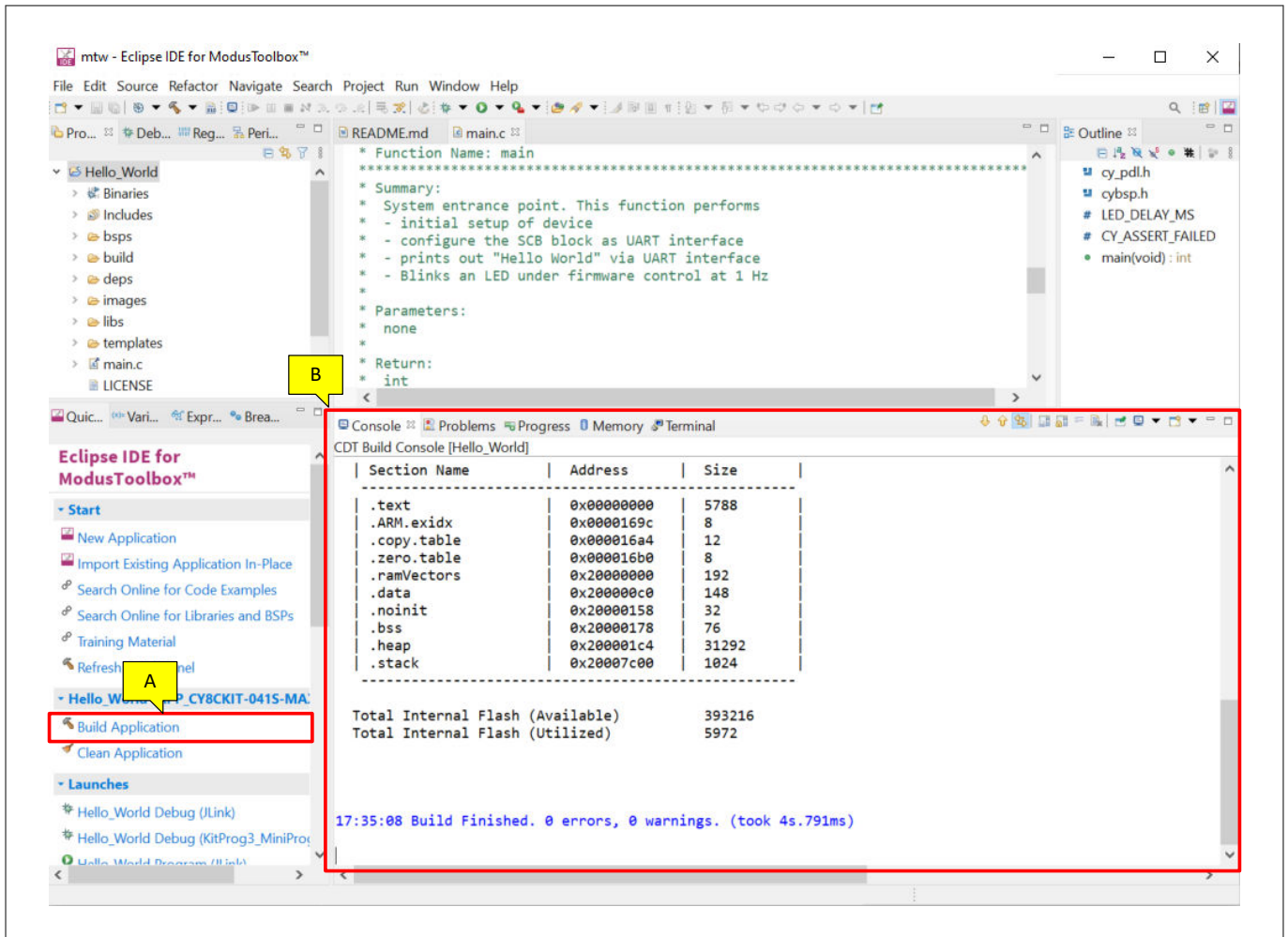


図 26 アプリケーションのビルド

エラーがある場合は、手順をチェックして、必要なタスクがすべて完了していることを確認してください。

注: CLI を使用してもアプリケーションをビルドできます。ModusToolbox™ user guide の 3.5.3 Use command line セクションを参照してください。このドキュメントは、ModusToolbox™ インストールディレクトリの /ide_3.2/docs/ フォルダにあります。

5 はじめての PSOC™ 4 設計

5.9 デバイスのプログラム

ここでは、PSOC™ 4 デバイスのプログラム方法を説明します。

ModusToolbox™は、SWD プロトコルを使用して、PSOC™ 4 デバイス上のアプリケーションをプログラムおよびデバッグします。ModusToolbox™がキット上のデバイスを識別するためには、キットで KitProg3 が実行されている必要があります。一部のキットには、KitProg3 ではなく KitProg2 ファームウェアが付属します。ModusToolbox™には、KitProg ファームウェアを KitProg2 から KitProg3 に切り替えるための fw-loader コマンドラインツールが含まれます。詳細は、[ModusToolbox™ IDE user guide](#) の 5.1.10 KitProg Firmware Loader セクションを参照してください。

自分でハードウェアを開発している場合は、ハードウェアプログラマー/デバッガが必要になる場合があります。例えば、[CY8CKIT-005 MiniProg4](#) です。

1. キットをユーザのコンピュータの USB ポートに接続してください。
2. アプリケーションプロジェクトを選択し、クイックパネルの **Launches** グループの下にある **<application name> Program (KitProg3_MiniProg4)** ショートカットをクリックしてください。IDE は、適切な実行構成を選択して実行します。最後のビルド以降にファイルが変更されている場合、このステップでもビルドが実行されることに注意してください。

5 はじめての PSOC™ 4 設計

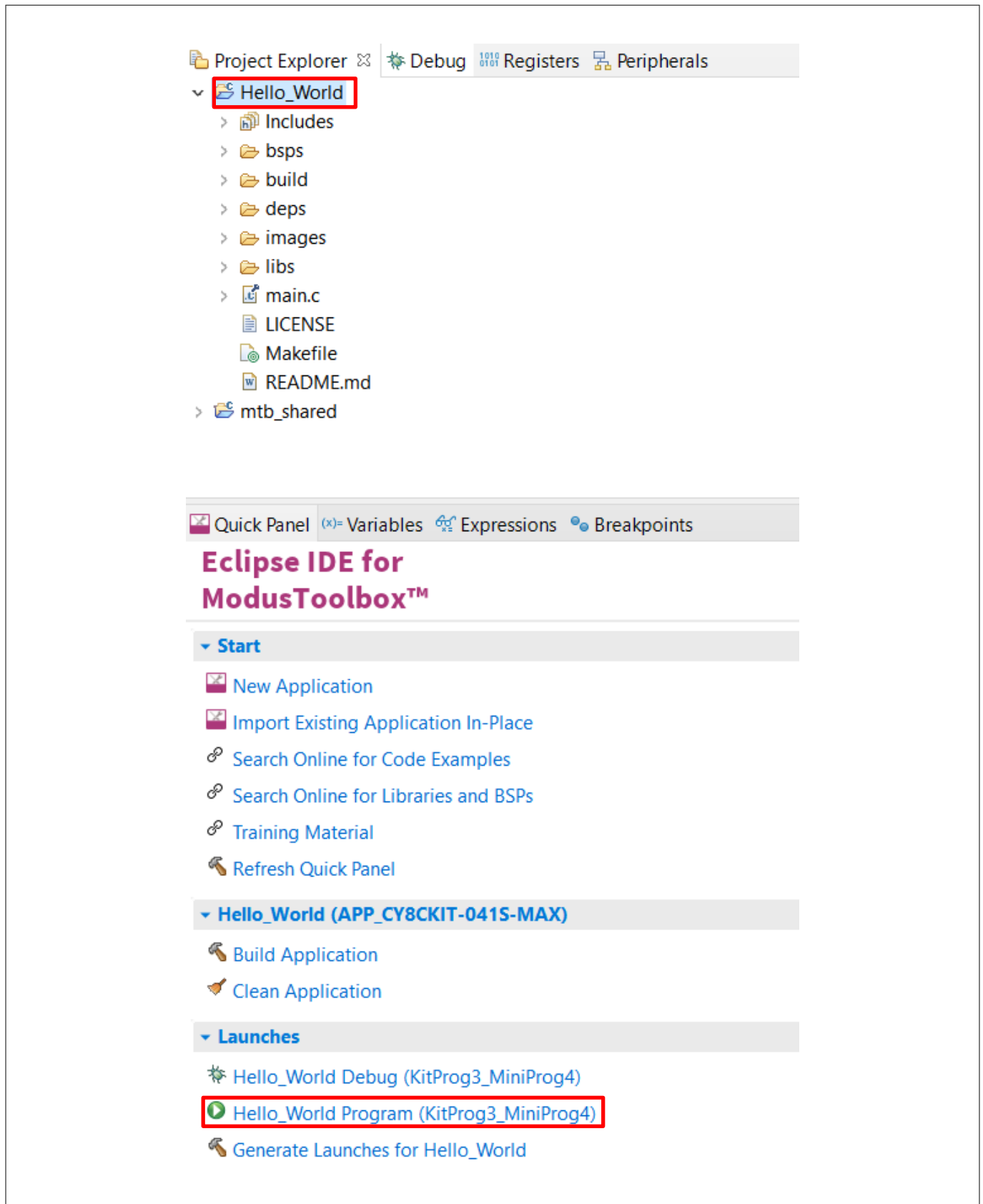


図 27 デバイスへのアプリケーションのプログラム

コンソールビューには、プログラミング操作の結果が一覧表示されます。

5 はじめての PSOC™ 4 設計



図 28 コンソール - プログラミング結果

5.10 設計のテスト

ここでは、設計をテストする方法について説明します。

プログラミングが完了したら、キットをリセットし、LED の点滅を確認します。

UART が有効になっているキットの場合は、次の手順に従って「Hello world」が印刷されることを確認します。このアプリケーションノートでは、Tera Term を UART ターミナルエミュレータとして使用して結果を表示します。任意の端末を使用して出力を表示できます。CY8CKIT-040T キットでは UART がサポートされていないため、これらの手順をスキップしてください。

1. シリアルポートの選択

図 29 に示すように、Tera Term を起動し、USB-UART COM ポートを選択してください。COM ポート番号は異なる場合があることに注意してください。

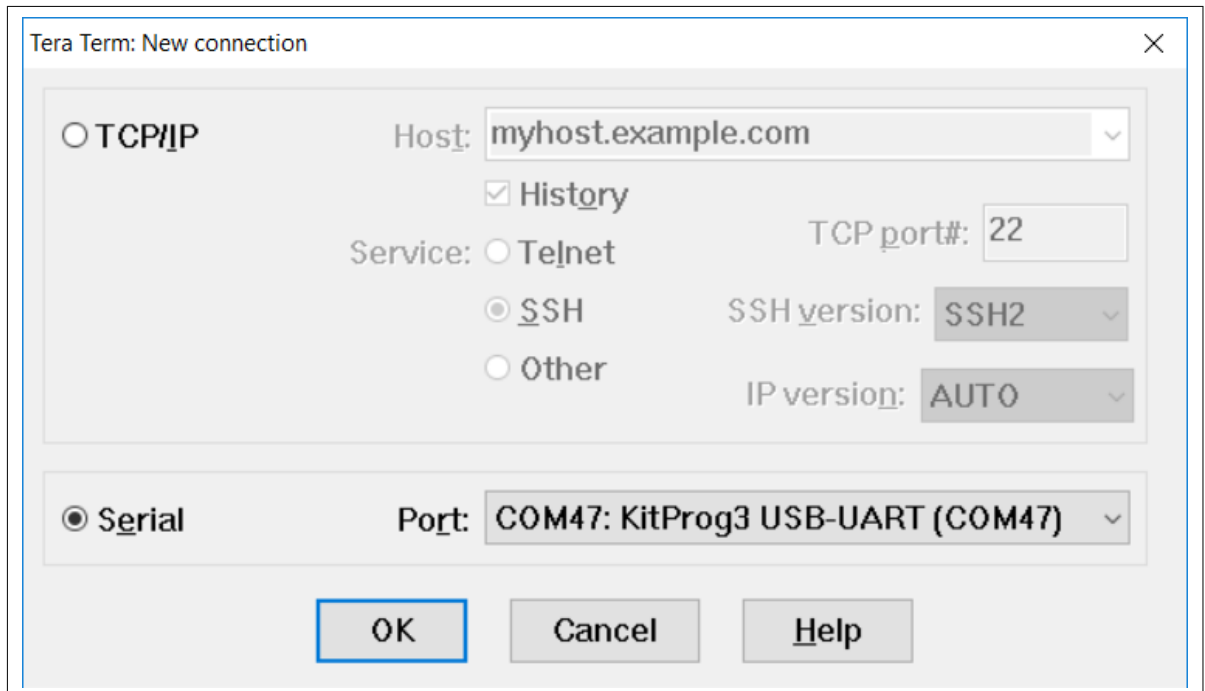


図 29 Tera Term での KitProg3 COM ポートの選択

2. ボーレートの設定

Setup > Serial port へ移動してください。ボーレートを 115200 に設定してください。

5 はじめての PSOC™ 4 設計

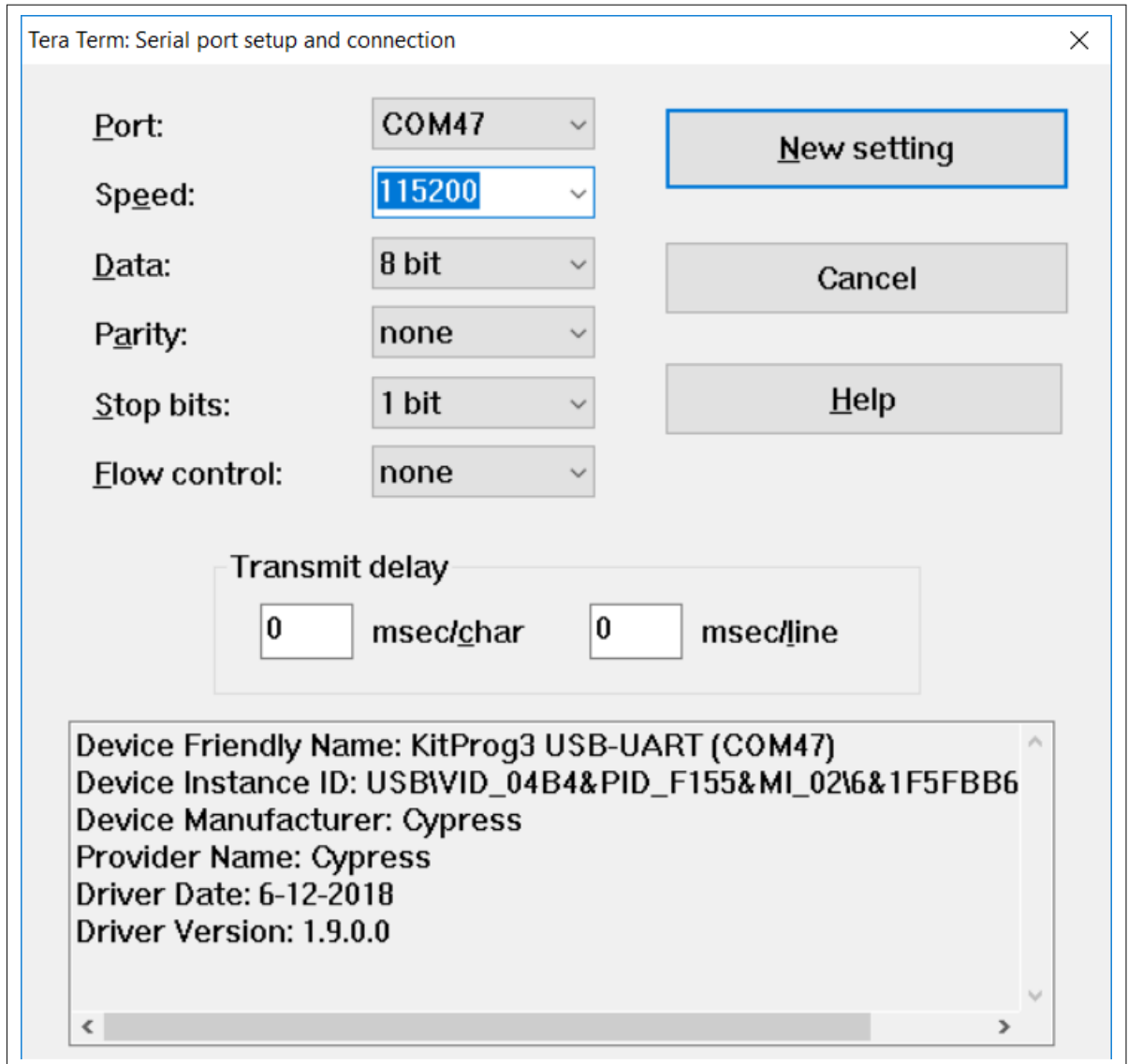


図 30 Tera Term でのボーレートの設定

3. デバイスのリセット

PSOC™ 4 キットのリセットスイッチを押してください (表 9 を参照してください)。「Hello World」メッセージが端末に表示されます。キットのユーザー LED が点滅し始めます。

表 9 PSOC™ 4 キット全体のリセットスイッチのピンマッピング表

機能	CY8CKIT-145 (PSOC™ 4000S)	CY8CKIT-149 (PSOC™ 4100S Plus)	CY8CKIT-041S-MAX (PSOC™ 4100S Max)	CY8CKIT-040T PSOC™ 4000T CAPSENS E™ Evaluation Kit	CY8CPROTO-040T PSOC™ 4000T CAPSENSE™ Prototyping Kit	CY8CPROTO-041T-PLUS CAPSENS E™ Prototyping Kit
RESET	SW1	SW2	SW1	SW1	SW1	SW1

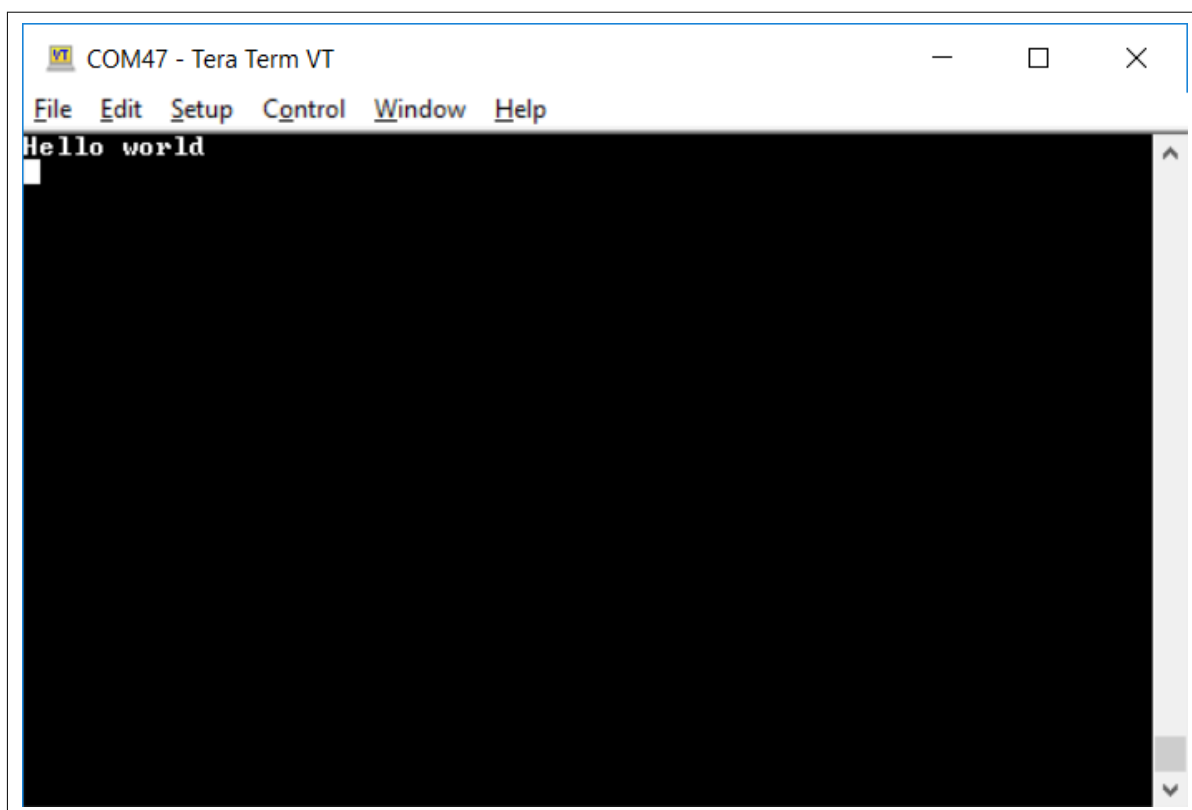


図 31 PSOC™ 4 からプリントされた UART メッセージ

6 PSOC™ Creator を使用するはじめての PSOC™ 4 設計

6 PSOC™ Creator を使用するはじめての PSOC™ 4 設計

ここでは以下を説明します。

- 従来の MCU 以上のことができるように PSOC™ をプログラムするデモンストレーション
- 単純な PSOC™ の設計をビルドし、それを開発キットにインストールする方法の提示
- PSOC™ 設計技術の学習を容易にする詳細な手順と、PSOC™ Creator の使用方法の提示

6.1 インストールの前に

6.1.1 PSOC™ Creator をインストールしましたか？

PSOC™ Creator ホームページから PSOC™ Creator をダウンロードし、インストールします。ツールセットのインストールは時間がかかることに注意してください。詳細については PSOC™ Creator リリース ノートを参照してください。

6.1.2 開発キットまたは Prototyping Kit をお持ちですか？

この設計をテストするためには、プログラマが統合された、表 10 にリストされているキットの 1 つが必要です。

表 10 PSOC™ 4 Pioneer Kit, Prototyping Kit および対応するデバイスの一覧

キット名	キット形式	対応するデバイスファミリ	製品番号
CY8CKIT-040	Pioneer kit	PSOC™ 4000	CY8C4014LQI-422
CY8CKIT-042	Pioneer kit	PSOC™ 4200	CY8C4245AXI-483
CY8CKIT-044	Pioneer kit	PSOC™ 4200M	CY8C4247AZI-M485
CY8CKIT-046	Pioneer kit	PSOC™ 4200L	CY8C4248BZI-L489
CY8CKIT-042-BLE	Pioneer kit	PSOC™ 4200 Bluetooth® LE	CY8C4247LQI-BL483
CY8CKIT-045S	Pioneer kit	PSOC™ 4500S	CY8C4548AZI-S485
CY8CKIT-043	Prototyping kit	PSOC™ 4200M	CY8C4247AZI-M485
CY8CKIT-145	Prototyping kit	PSOC™ 4000S	CY8C4045AZI-S413
CY8CKIT-147	Prototyping kit	PSOC™ 4100PS	CY8C4145LQI-PS433
CY8CKIT-149	Prototyping kit	PSOC™ 4100S Plus	CY8C4147AZI-S475

6.1.3 実行中のプロジェクトを参照したいですか？

設計プロセスを実行したくない場合は、PSOC™ Creator で **Find Code Example (File > Code Example... > CE230991_My_First_Project)** を使用して、完成した PSOC™ Creator プロジェクトを取得できます。その後、**ビルド**と**プログラムの**ステップにジャンプできます。

6.2 設計について

図 32 に示すように、この設計は TCPWM コンポーネントを使用し、単純に 2 個の LED を点滅させます。TCPWM は PWM モードとして設定されます。PWM の 2 個の相補出力は LED を制御します。PWM は LED のトグルが見えるように、非常に低い周波数と 50 パーセントのデューティ比で動作します。2 個の個別の LED の代わりにデュアルカラー LED を使用すると、このプロジェクトはデュアルカラー LED の色をトグルできます。

6 PSOC™ Creator を使用するはじめての PSOC™ 4 設計

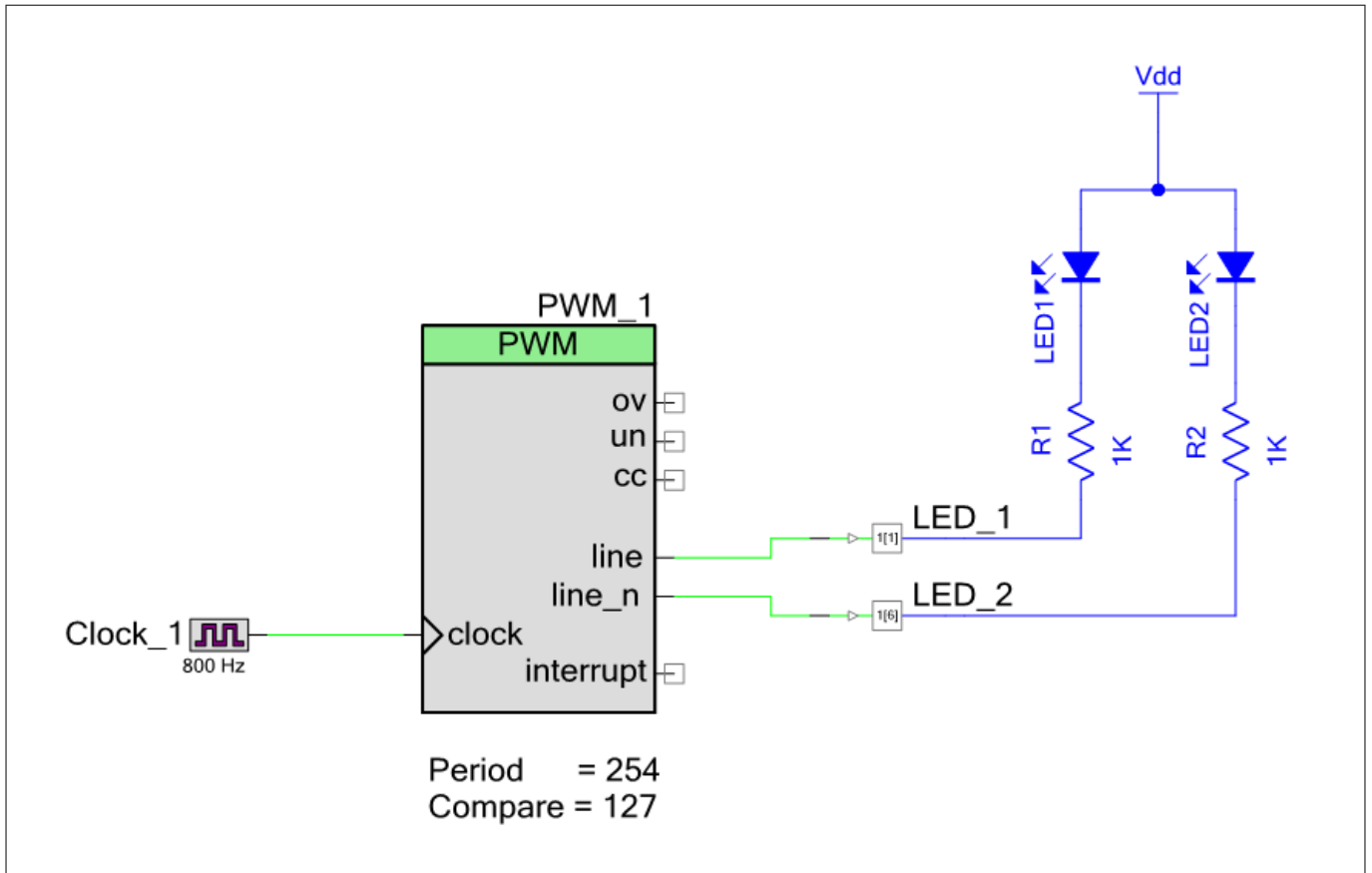


図 32 初めての PSOC™ 4 設計

6.3 パート 1: 設計の作成

ここでは設計プロセスの概要について順に説明します。空のプロジェクトを作成し、ハードウェアとファームウェア設計入力について解説します。

1. 図 33 に示すように、PSOC™ Creator を起動し、File メニューから **New > Project** を選択してください。

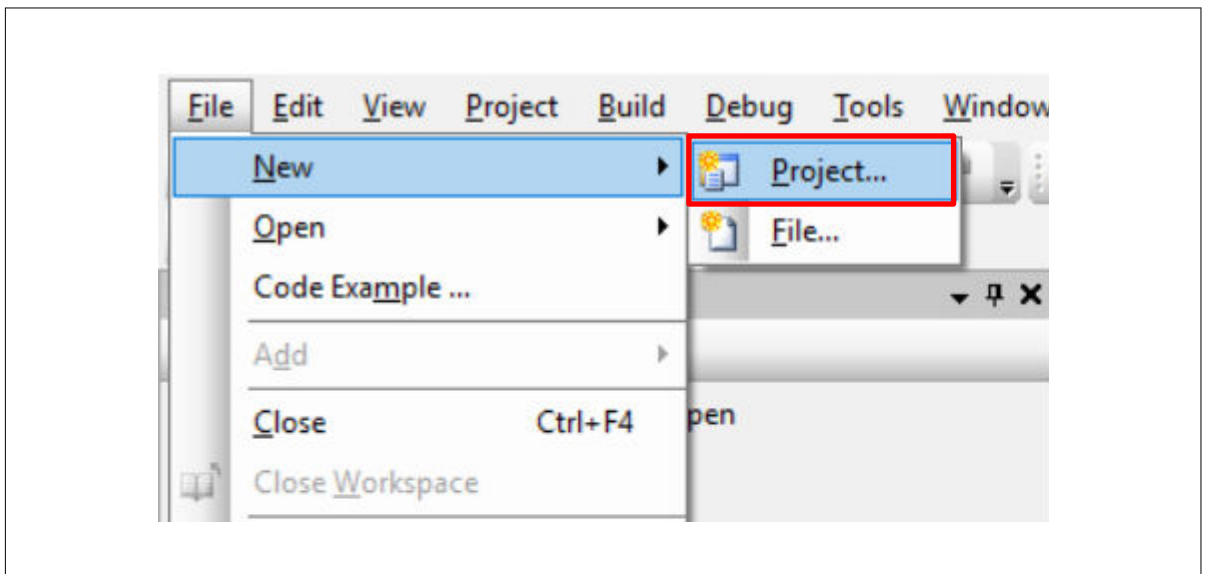


図 33 新規プロジェクトの作成

6 PSOC™ Creator を使用するはじめての PSOC™ 4 設計

2. ポップアップ ウィンドウで開発キットを選択してください。例えば、CY8CKIT-149 をお持ちの場合、**Kit: CY8CKIT-149 (PSOC™ 4100S Plus)**を選択し、**Next** をクリックしてください。メニューにお持ちの PSOC™ 4 開発キットがない場合、ウェブサイトからキット セットアップをダウンロードし、インストールしてください。または、ターゲット ハードウェアの代わりに、ターゲット デバイス ラジオ ボタンを選択して適切なデバイスを選んで、**Next** をクリックすることもできます。

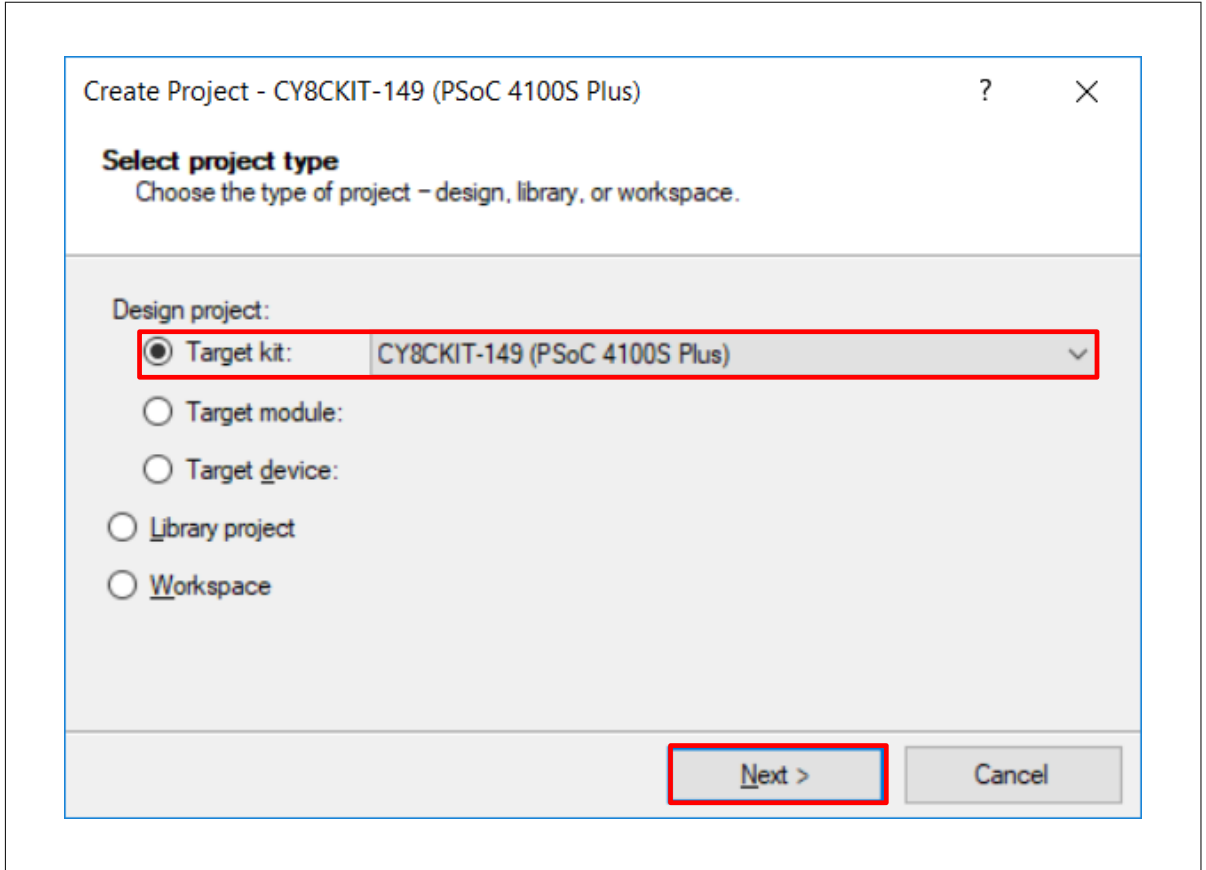


図 34 空の PSOC™ 4 新規プロジェクトの作成

3. 次のウィンドウから **Empty Schematic** オプションを選択し、**Next** をクリックしてください。

6 PSOC™ Creator を使用するはじめての PSOC™ 4 設計

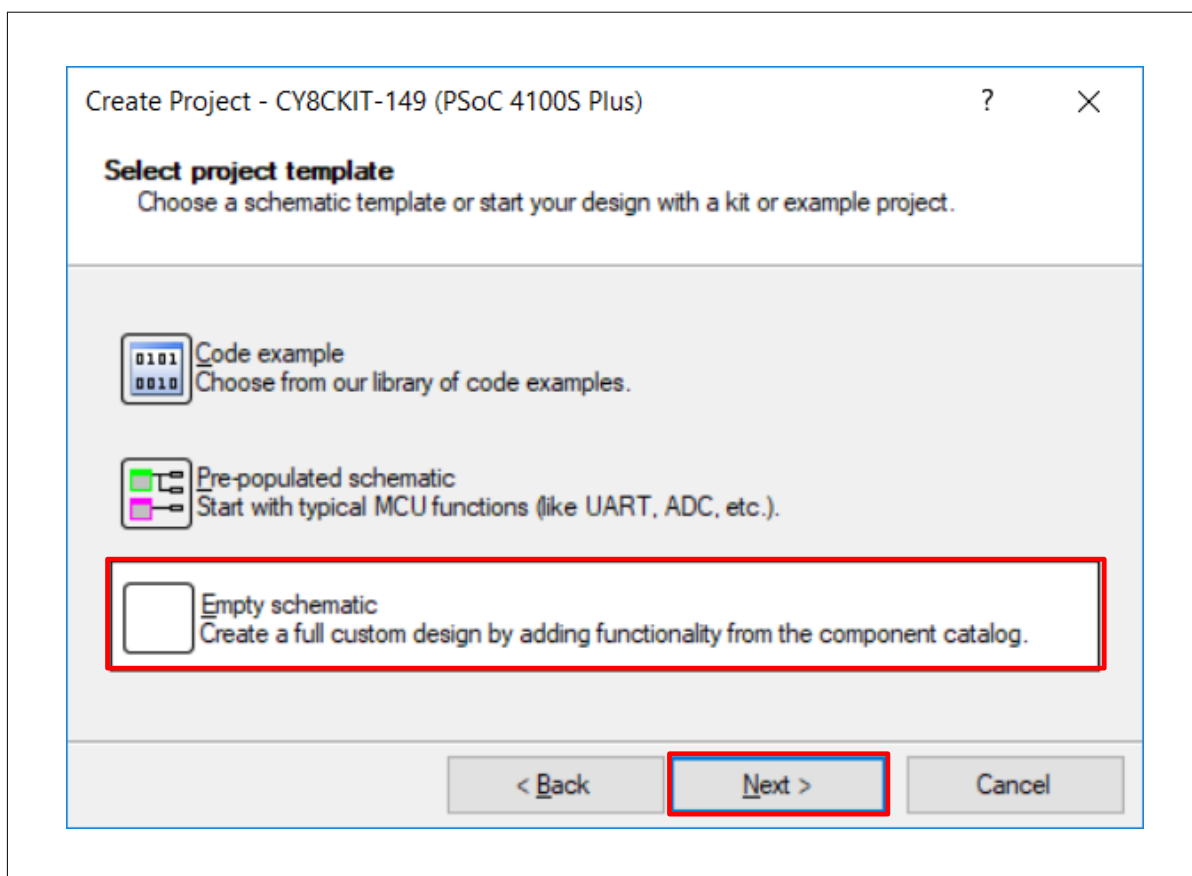


図 35 Empty schematic の選択

4. [図 36](#) に示すように、プロジェクト名 (例えば「My_First_Project」) およびワークスペース名をつけてください。新規プロジェクトに適切な場所を選択して、**Finish** をクリックしてください。

6 PSOC™ Creator を使用するはじめての PSOC™ 4 設計

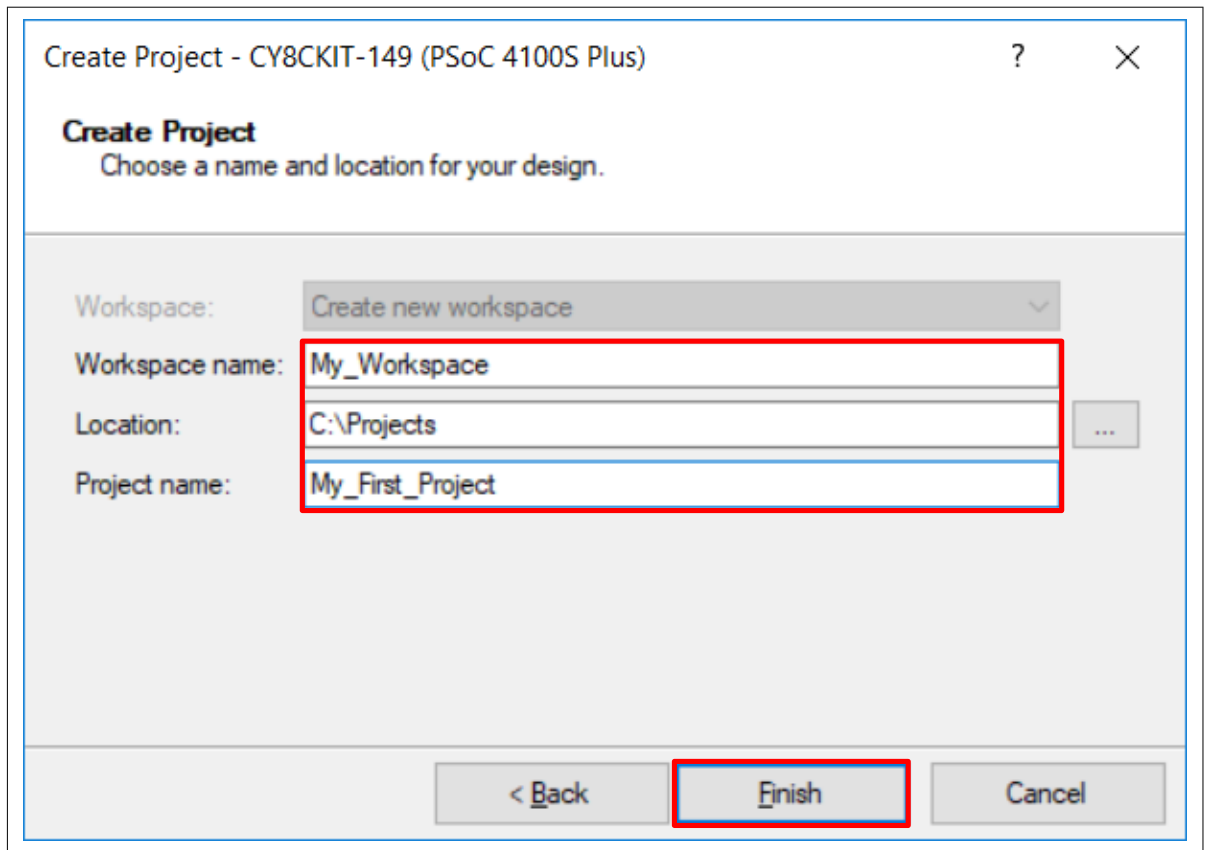


図 36 プロジェクト名と場所の選択

5. **Workspace Explorer** (図 37 を参照) に示すように、新規プロジェクトを作成すると、ファイルのベースラインセットを含むプロジェクトフォルダが作成されます。**TopDesign.cysch** をダブルクリックし、プロジェクトの回路図ファイルを開いてください。

6 PSOC™ Creator を使用するはじめての PSOC™ 4 設計

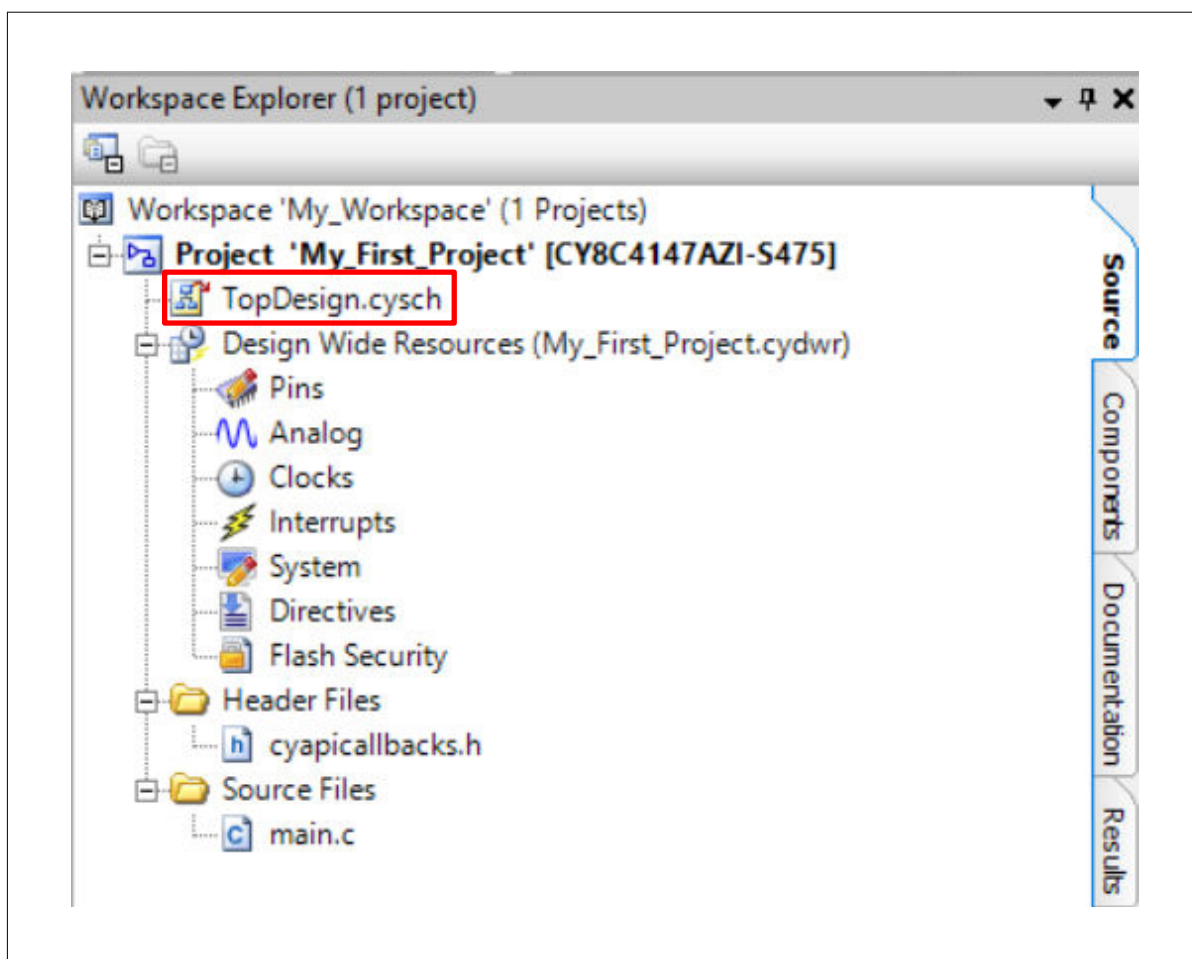


図 37 TopDesign 回路図を開く

6. [図 38](#) に示すように、Component Catalog から回路図に 1 つの PWM (TCPWM mode) コンポーネントをドラッグしてください。

6 PSOC™ Creator を使用するはじめての PSOC™ 4 設計

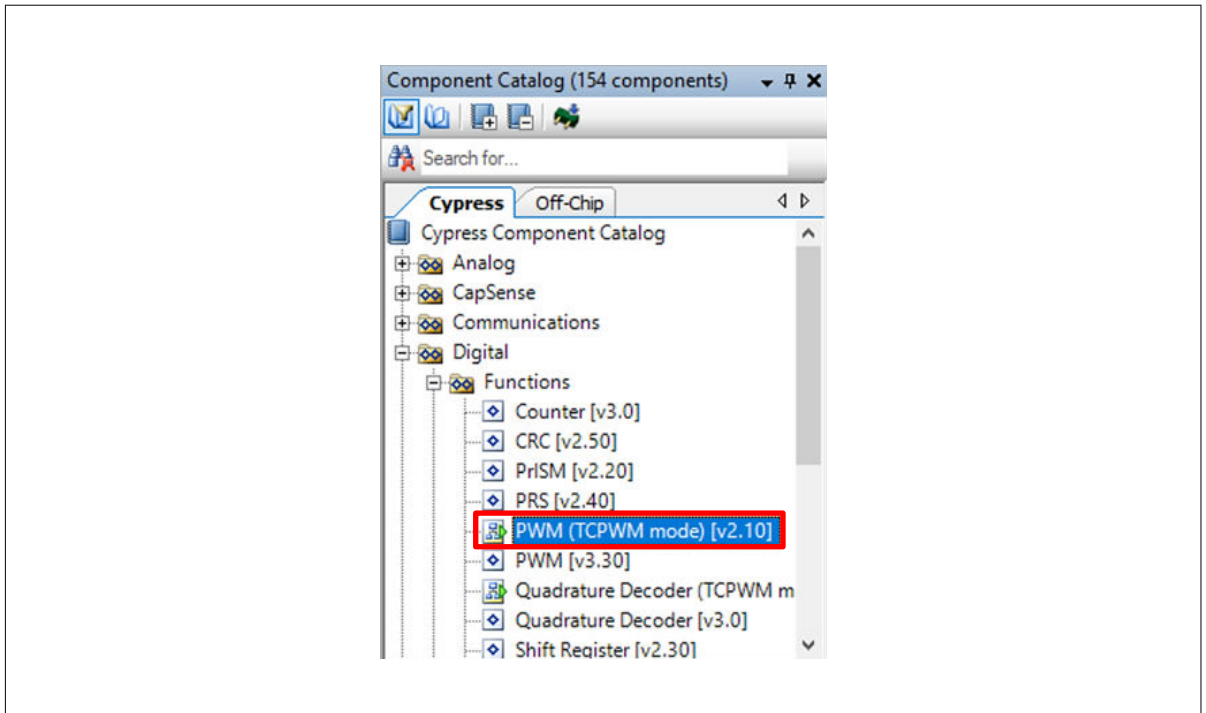


図 38 PWM コンポーネントの配置

7. 図 39 に示すように、回路図の PWM コンポーネントをダブルクリックし、コンポーネントのプロパティを設定してください。**PWM** タブをクリックして、デューティ比が 50 パーセントの PWM 信号を生成するために、周期値を 254 に**比較値**を 127 にセットしてください。
入力クロック周波数を 8 で分周するために、**Prescaler** を 8x にセットしてください。

6 PSOC™ Creator を使用するはじめての PSOC™ 4 設計

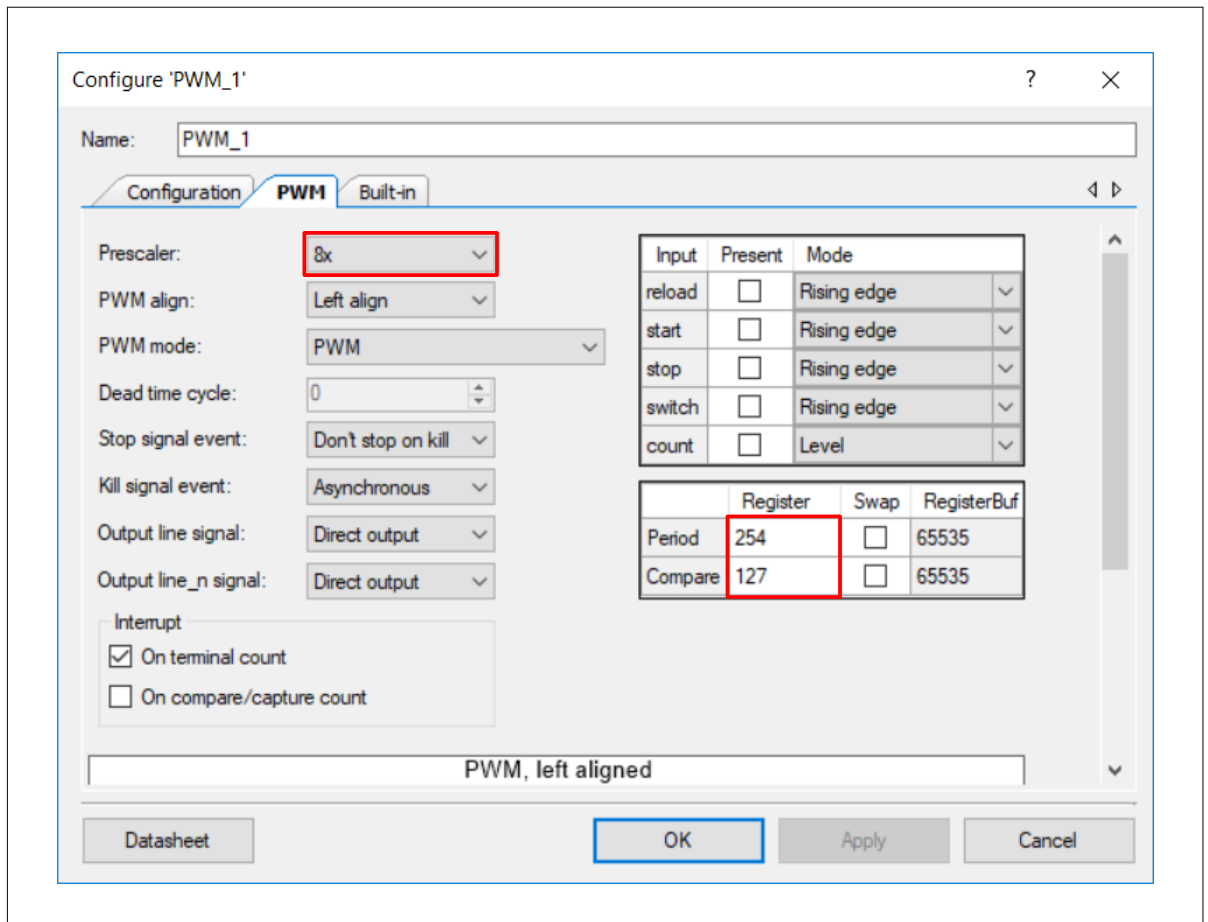


図 39 PWM コンポーネントの設定

8. PWM コンポーネントが動作するために 1 つの入カクロックを必要とします。図 40 と図 41 に示すように、回路図に **Clock** コンポーネントをドラッグ アンドドロップし、そのコンポーネントをダブルクリックして、**Frequency** を 800 Hz に設定してください。PWM コンポーネントの Prescaler 値設定が 8 なので、PWM の実効入カクロックはわずか 100 Hz です。したがって、PWM 周期が 254 の場合、PWM 出力周期時間は 2.54 秒になります。

6 PSOC™ Creator を使用するはじめての PSOC™ 4 設計

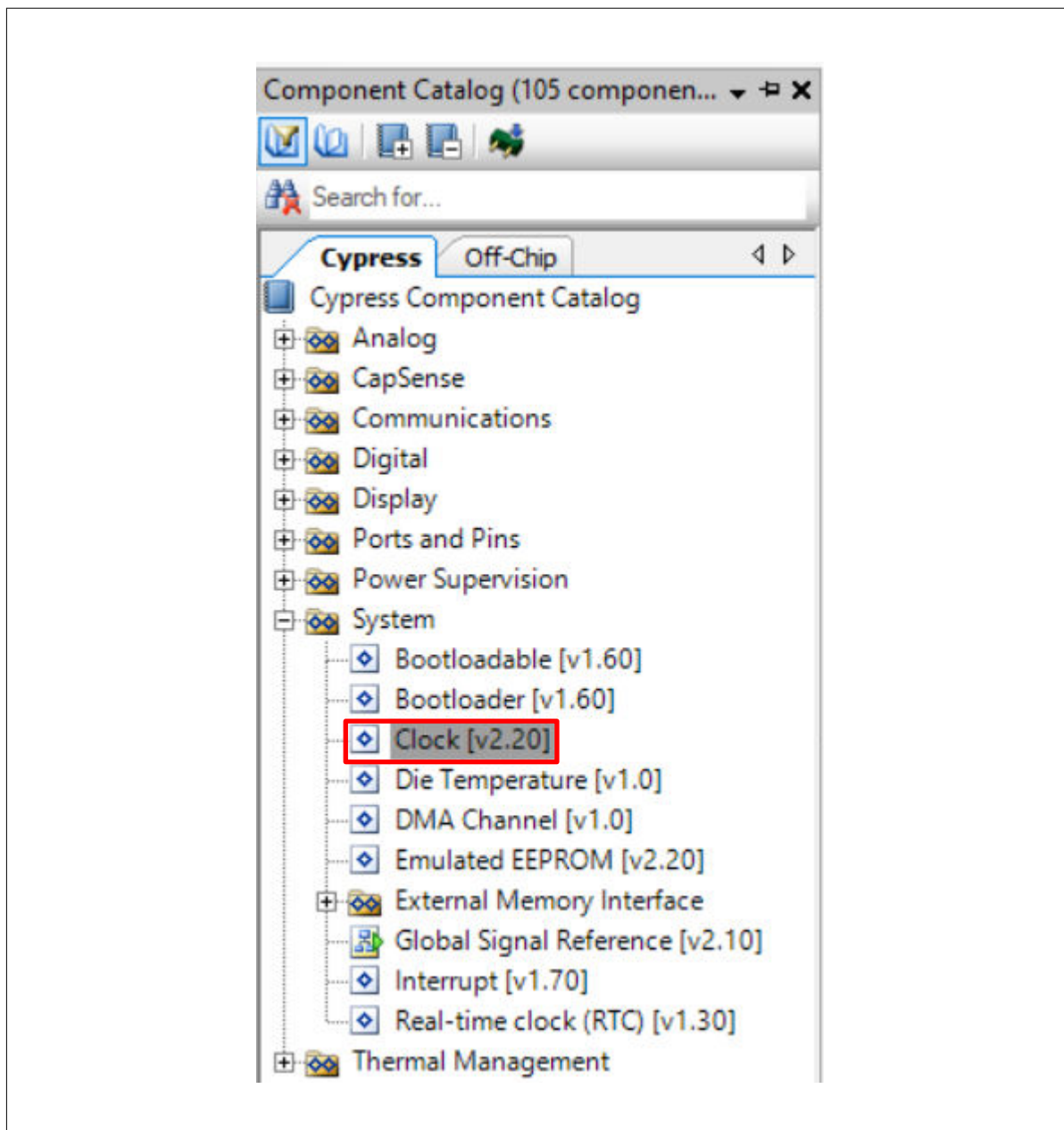


図 40 クロックコンポーネントの場所

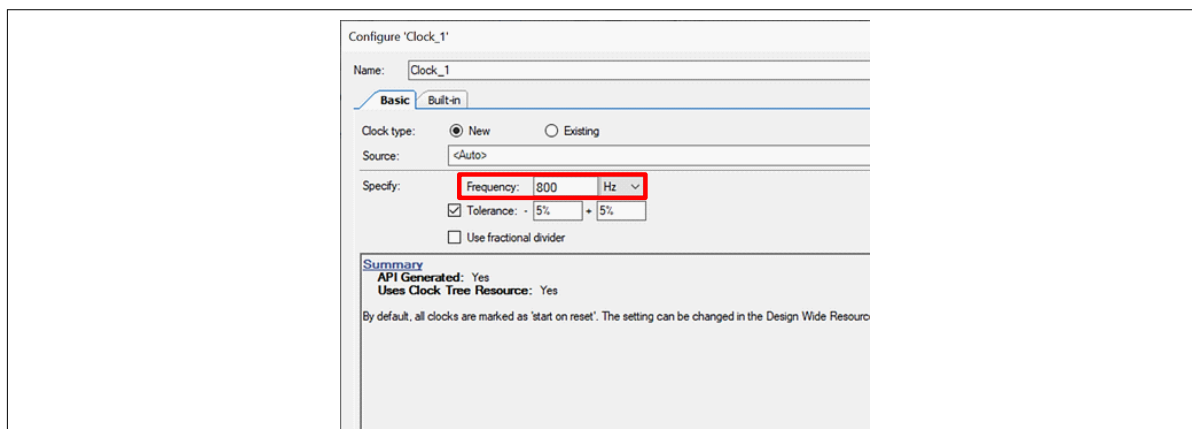


図 41 クロックコンポーネントの設定

6 PSOC™ Creator を使用するはじめての PSOC™ 4 設計

9. **Digital Output Pin** コンポーネントをドラッグ アンドドロップしてください。図 42 と図 43 に示すように、その名前を LED_1 に変更してください。他のデジタル出力ピンコンポーネントを追加して、LED_2 と名付けてください。

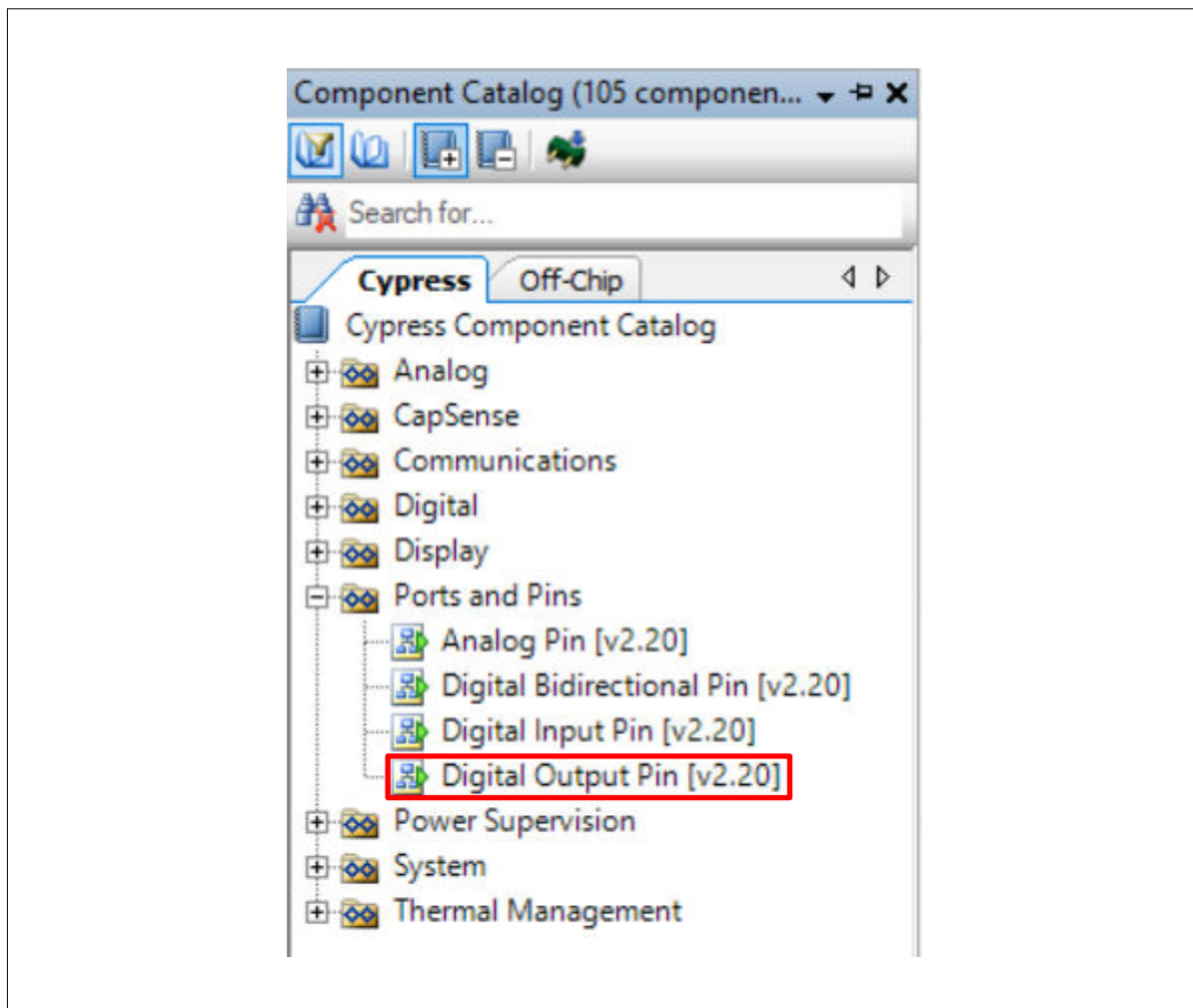


図 42 デジタル出力ピンコンポーネントの場所

6 PSOC™ Creator を使用するはじめての PSOC™ 4 設計

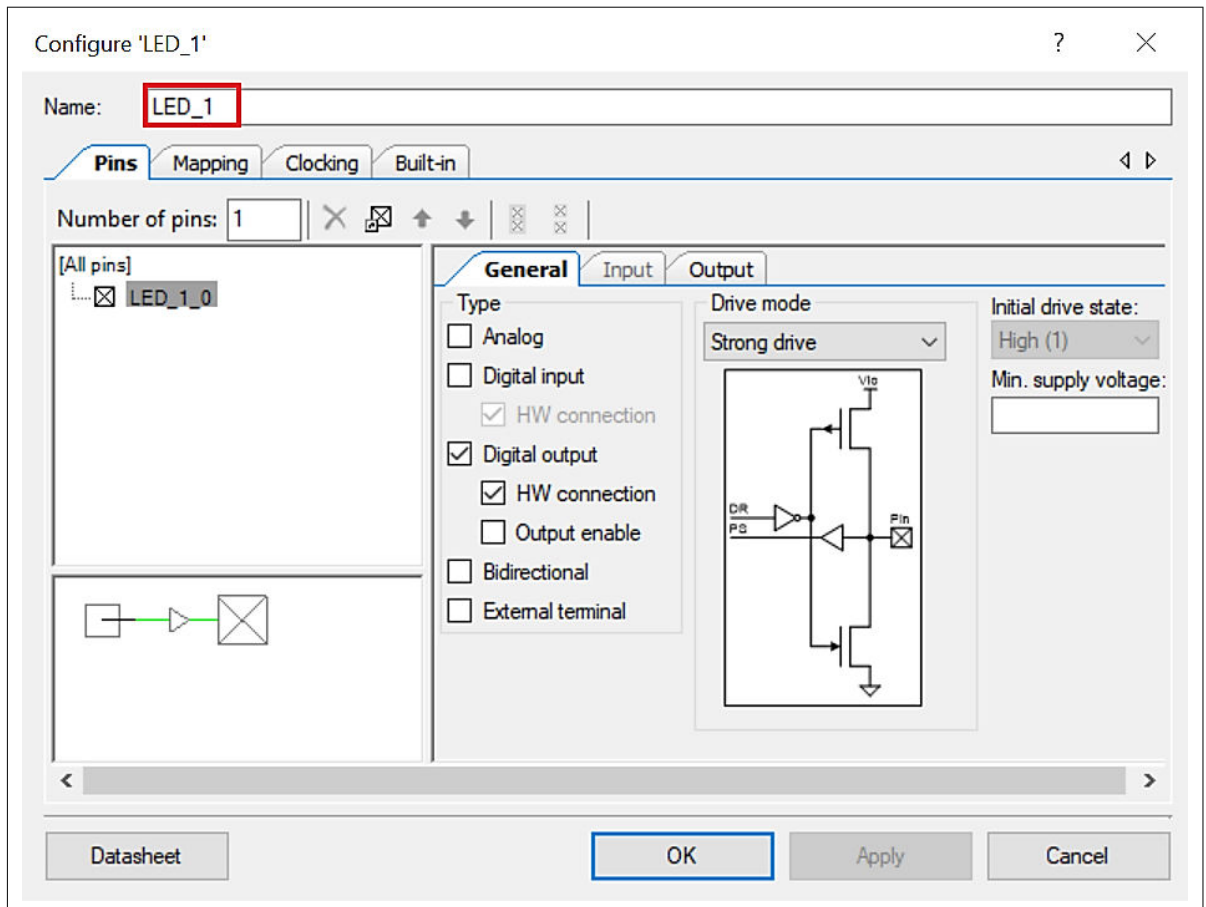


図 43 ピンコンポーネントの再名付け

10. 回路図ウィンドウで、図 44 に示すようにワイヤー ツールを選択、または **W** を押してください。

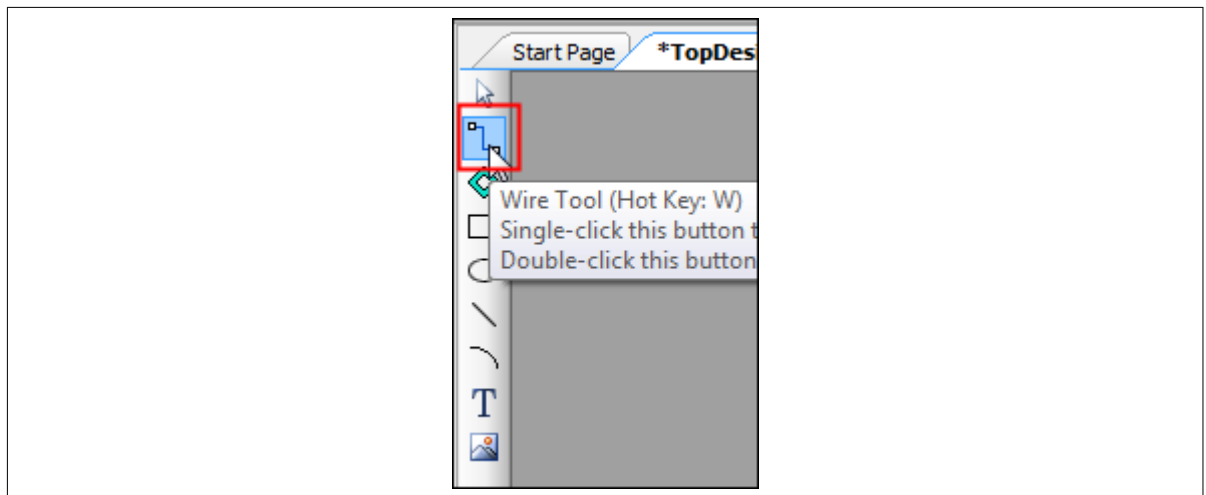


図 44 ワイヤーツールの選択

11. 図 45 に示すように、コンポーネントを一緒に配線してください。

6 PSOC™ Creator を使用するはじめての PSOC™ 4 設計

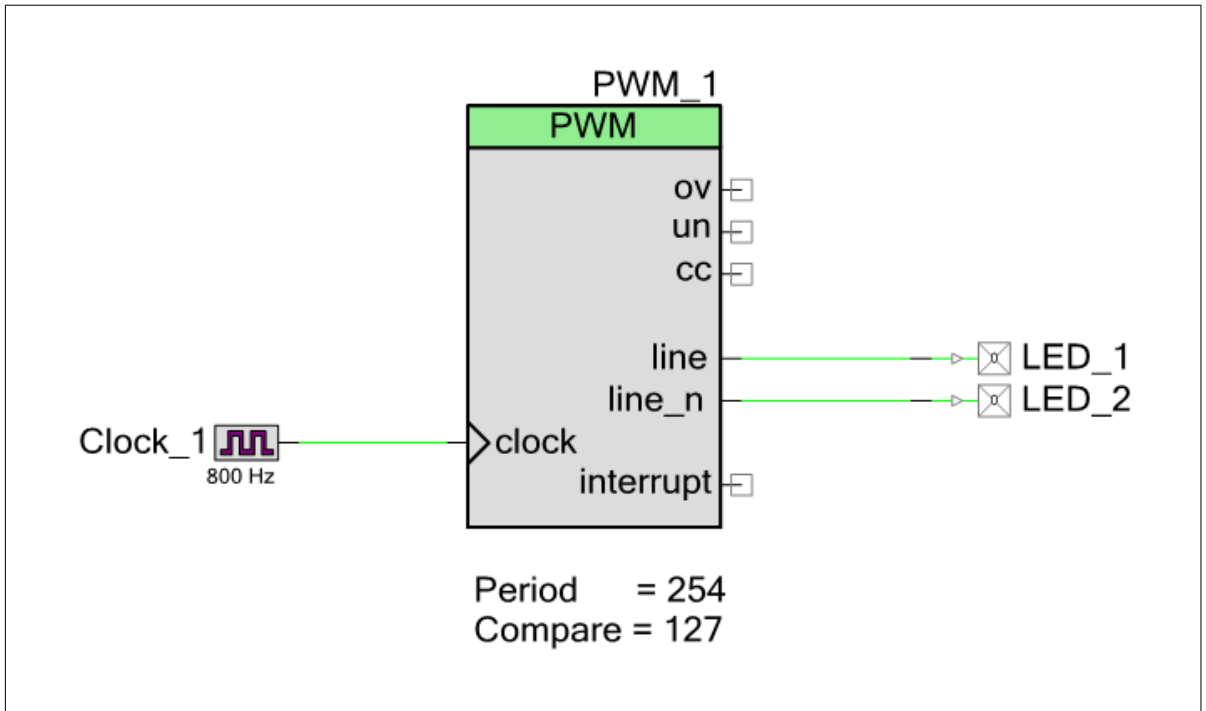


図 45 回路図の配線

12. ほとんどのコンポーネントはデバイスリセットで無効になります。クロックコンポーネントは重要な例外で、デフォルトで自動的に起動されます。これらのコンポーネントを有効にするように、プロジェクトにコードを追加する必要があります。Workspace Explorer から main.c を開き、Code Listing 3 のように、コードを main()関数に追加してください。

Code Listing 3 PWM コンポーネントを有効化

```
int main(void)
{
    /* Enable and start the PWM */
    PWM_1_Start();

    for(;;)
    {

    }
}
```

13. ビルドメニューから **Build My_First_Project** を選択してください。図 46 に示すように、Workspace Explorer ウィンドウでは、PSOC™ Creator が PWM, クロック, およびデジタル出力ピンコンポーネントに対しソースコードファイルを自動的に生成することに注意してください。

6 PSOC™ Creator を使用するはじめての PSOC™ 4 設計

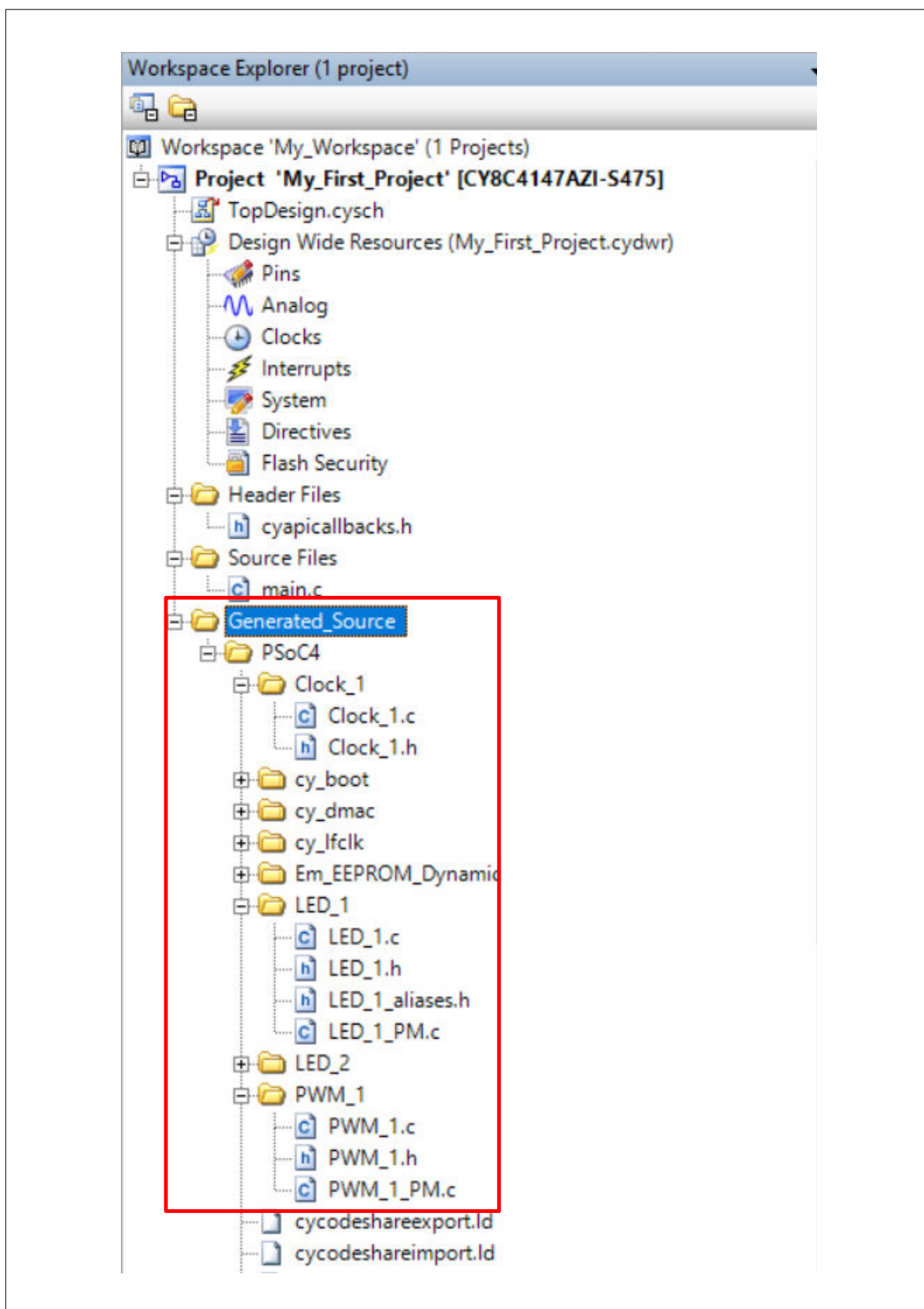


図 46 生成されたソース ファイル

14. **Workspace Explorer** から My_First_Project.cydwr (Design-Wide Resource ファイル) ファイルを開き、**Pins** タブをクリックしてください。このタブで LED_1 と LED_2 出力用のデバイスピンを選択できます。

6 PSOC™ Creator を使用するはじめての PSOC™ 4 設計

図 47 に、CY8CKIT-149 PSOC™ 4 prototyping kit で LED_1 と LED_2 ピンを LED に接続するピン コンフィギュレーションを示します。別の PSOC™ 4 pioneer kit をお使いの場合は表 11 を、PSOC™ 4 prototyping kit をお使いの場合は表 12 を参照してください。

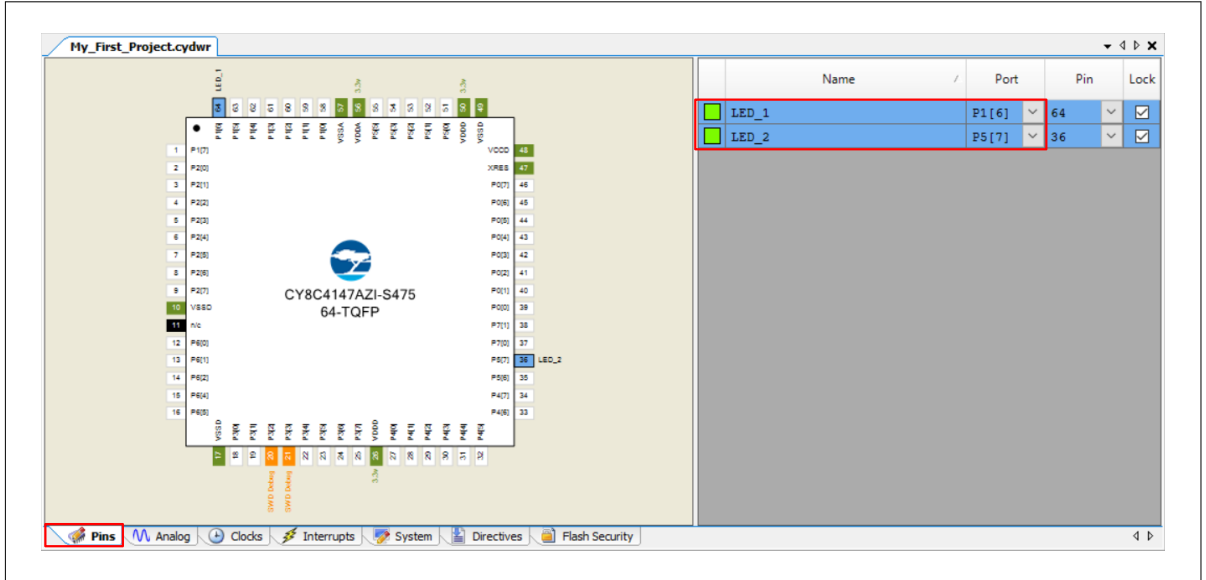


図 47 ピン選択

表 11 Pioneer キットのピン マッピング表

機能	CY8CKIT-040 (PSOC™ 4000)	CY8CKIT-041 (PSOC™ 4100S)	CY8CKIT-042 (PSOC™ 4200)	CY8CKIT-042- BLE (PSOC™ 4200 BLE)	CY8CKIT-044 (PSOC™ 4200M)	CY8CKIT-046 (PSOC™ 4200L)	CY8CKIT-045S (PSOC™ 4500S)
緑色 LED (アクティブ LOW)	P1[1]	P2[6]	P0[2]	P3[6]	P2[6]	P5[3]	P0[0]
赤色 LED (アクティブ LOW)	P3[2] ¹⁾	P3[4] ²⁾	P1[6]	P2[6]	P0[6]	P5[2]	P1[6]

- 1) PSOC™ 4000 製品には、補完的な PWM 出力用の固定ピン P1[1]および P1[6]があります。PWM 出力に他のピンは使用できません。詳細については、デバイスのデータシートを参照してください。CY8CKIT-040 を使用している場合は、P1[1]に接続する緑色の LED を LED1 として使用できます。赤色 LED を LED2 として使用するためには、ヘッダー J4 の P3[2]をヘッダー J3 の P1[6]にワイヤーで接続します。または、外部 LED も LED2 として P1[6]に接続できます。
- 2) 上記の注記と同様に、CY8CKIT-041 を使用している場合は、P2[6]に接続する緑色の LED を LED1 として使用し、LED2 の補完的な PWM 出力 P2[7]を使用できます。赤色 LED を LED2 として使用するためには、ヘッダー J2 の P3[4]をヘッダー J3 の P2[7]にワイヤーで接続します。または、外部 LED も LED2 として P2[7]に接続できます。

6 PSOC™ Creator を使用するはじめての PSOC™ 4 設計

表 12 Prototyping キットのピン マッピング表

機能	CY8CKIT-145 (PSOC™ 4000S)	CY8CKIT-149 (PSOC™ 4100S Plus)
緑色 LED - LED 1 (アクティブ LOW)	P3[4]	P1[6]
緑色 LED - LED 2 (アクティブ LOW)	P3[5]	P5[7]

- 最後に、[ステップ 13](#) で説明したようにプロジェクトを再ビルドします。

6.4 パート 2: デバイスのプログラム

ここでは、デバイスのプログラム方法を説明します。USB ケーブルを使ってキット ボードをコンピューターに接続してください。

- [図 48](#) に示すように、PSOC™ Creator メニューから、**Debug > Select Debug Target** を選択してください。

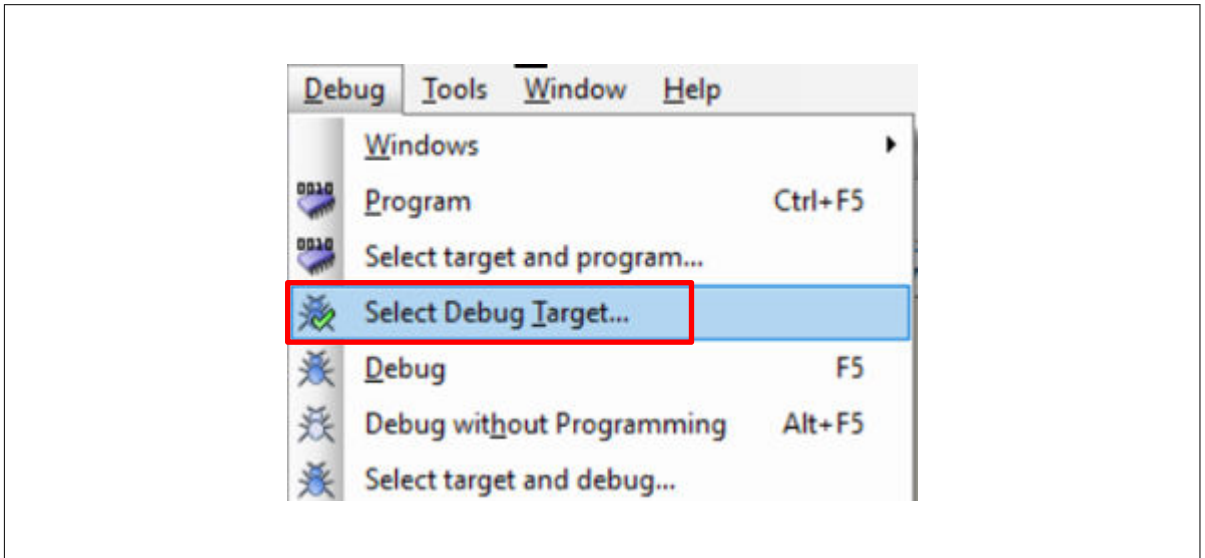


図 48 デバッグ ターゲットの選択

- [図 49](#) に示すように、**Select Debug Target** ダイアログボックスで、**Port Acquire** をクリックしてから、**Connect** をクリックしてください。OK をクリックし、ダイアログボックスを閉じてください。

6 PSOC™ Creator を使用するはじめての PSOC™ 4 設計

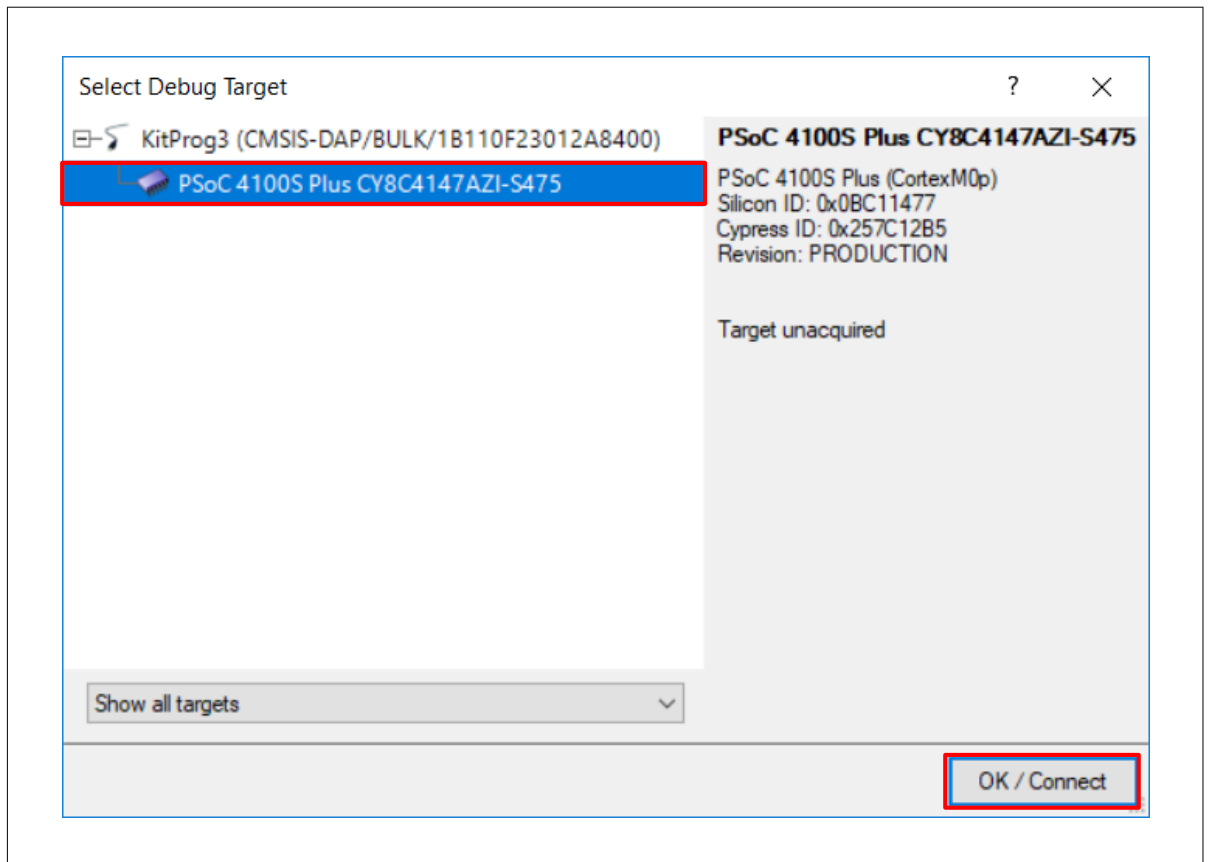


図 49 デバイスの接続

3. 図 50 に示すように、**Debug > Program** を選択し、プロジェクトでデバイスをプログラムしてください。

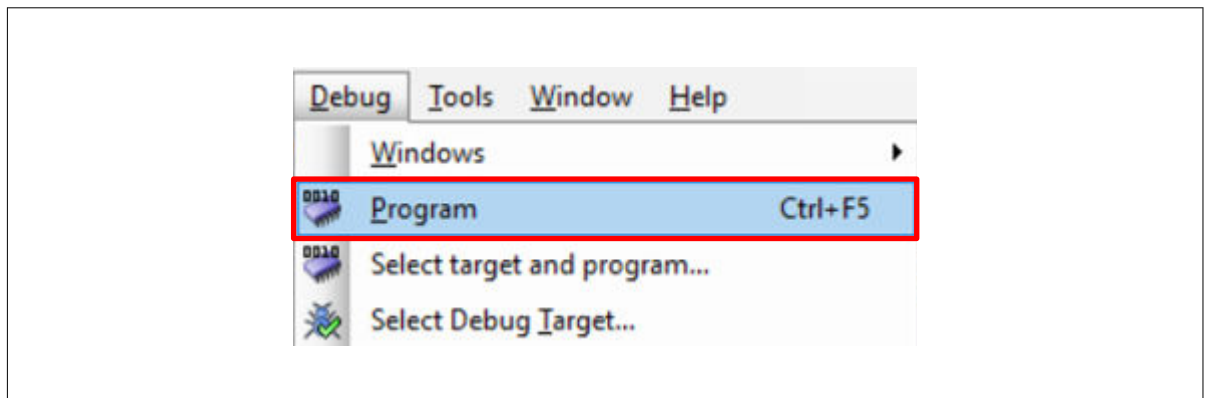


図 50 デバイスのプログラミング

4. 図 51 に示すように、プログラミングの状態はステータスバー (ウィンドウの左下隅) に表示されます。

6 PSOC™ Creator を使用するはじめての PSOC™ 4 設計

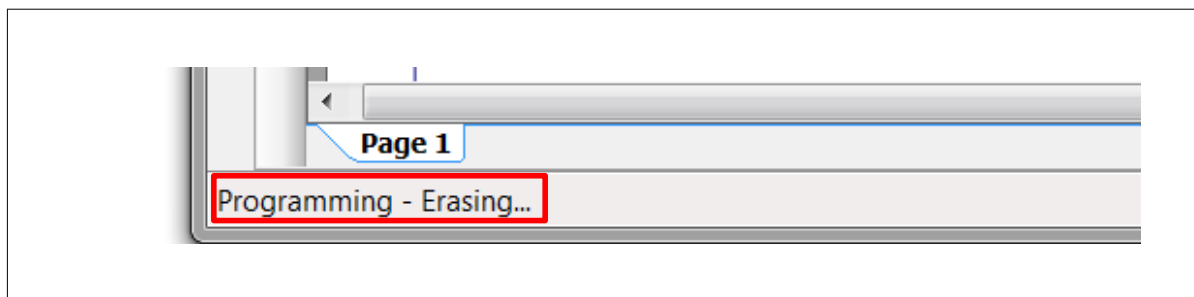


図 51 プログラミングの状態

5. デバイスがプログラムされた後、LED のトグルをチェックし、プロジェクトの動作を確認してください。

7 まとめ

7 まとめ

本アプリケーションノートは PSOC™ 4 アーキテクチャおよび開発ツールを説明しました。PSOC™ 4 は単一チップ上に設定可能なアナログとデジタルのペリフェラル機能、メモリ、および Arm® Cortex®-M0/M0+ 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組込みシステムオンチップです。内蔵された機能と低リーク電力モードにより、PSOC™ 4 は低消費電力とコスト効率が高い組込みシステムに理想的です。

また、本アプリケーションノートは PSOC™ 4 を迅速に深く理解できるように包括的なリソースをご紹介します。

参考資料

参考資料

- [1] [AN54181: PSOC™ 3 入門](#)
- [2] [AN77759: PSOC™ 5LP 入門](#)

改訂履歴

改訂履歴

版数	日付	変更内容
英語版(**)	-	この版は英語版のみです。英語版の改訂内容: New Application Note
英語版(*A)	-	この版は英語版のみです。英語版の改訂内容: Demo project changed to leverage Pioneer kit. Added architecture introduction.
**	2013-09-20	これは英語版 001-79953 Rev. *B を翻訳した日本語版 Rev. ** です。英語版の改訂内容: Reformatted graphics. Updated links.
英語版(*C)	-	この版は英語版のみです。英語版の改訂内容: Updated attached Associated Project files. Updated content across the entire document. Updated in new template.
英語版(*G)	-	この版は英語版のみです。英語版の改訂内容: Updated the projects and the respective section in the AN to support PSoC™ Creator 3.0 SP1 and PSoC™ 4000 device.
英語版(*H)	-	この版は英語版のみです。英語版の改訂内容: Added Code Examples section. Minor edits and format changes throughout the document.
*A	2015-04-17	これは英語版 001-79953 Rev. *I を翻訳した日本語版 Rev. *A です。英語版の改訂内容: Added More Information section. Removed detailed feature descriptions. Updated for PSoC™ 4200M family of devices.
英語版(*J)	-	この版は英語版のみです。英語版の改訂内容: Updated PSOC™リソース and PSOC™が MCU より優れている点 . Added the following sections: Convert Project to Bootloadable for CY8CKIT-049, Bootload Your CY8CKIT-049, and More PSoC™ 4 Code Examples. Updated 図 8 .
*B	2015-11-16	これは英語版 001-79953 Rev. *K を翻訳した日本語版 Rev. *B です。英語版の改訂内容: Updated for PSoC™ 4200L. Updated the example projects to PSoC™ Creator 3.3.
英語版(*L)	-	この版は英語版のみです。英語版の改訂内容: Updated the example projects to PSoC™ Creator 3.3 SP1.
*C	2016-02-23	これは英語版 001-79953 Rev. *M を翻訳した日本語版 Rev. *C です。英語版の改訂内容: Updated for PSoC™ 4000S and PSoC™ 4100S. Updated the example projects to PSoC™ Creator 3.3 SP2. Updated 表 10 , 表 11 to add PSoC™ 4200 BLE.
英語版(*N)	-	この版は英語版のみです。英語版の改訂内容: Updated logo and copyright
*F	2017-11-24	これは英語版 001-79953 Rev. *O を翻訳した日本語版 Rev. *F です。英語版の改訂内容: Added references to PSoC™ 4100S Plus throughout the document. Updated Table 1 to add PSoC™ 4100S Plus. Updated PSOC™リソース with the reference of AN64846 - Getting Started with CAPSENSE™ .

改訂履歴

版数	日付	変更内容
英語版(*P)	-	この版は英語版のみです。英語版の改訂内容: Updated the example projects to PSoC™ Creator 4.2 Updated Table 1 to add supported kit Updated 表 10 to add CY8CKIT-49, CY8CKIT-145 and CY8CKIT-149 Added 表 11 to include CY8CKIT-145 and CY8CKIT-149 pin mapping Updated Figure 7. Added example project of CY8CKIT-145 and CY8CKIT-149 as a part of the AN79953.zip file in this application note landing page.
英語版(*Q)	-	この版は英語版のみです。英語版の改訂内容: Updated template Updated for PSoC™ 4100PS Minor edits and format changes throughout the document Added example project of CY8CKIT-147 as a part of the AN79953.zip file
*G	2019-01-24	これは英語版 001-79953 Rev. *R を翻訳した日本語版 Rev. *G です。英語版の改訂内容: Updated template Corrected the link to PSoC™ 4100S Plus in PSOC™リソース
*H	2019-10-21	これは英語版 001-79953 Rev. *S を翻訳した日本語版 Rev. *H です。英語版の改訂内容: Updated Table 1 to add PSoC™ 4500 and update the specifications of PSoC™ 4100S Plus device family. Added Motor Control Accelerator feature.
*I	2021-01-25	これは英語版 001-79953 Rev. *T を翻訳した日本語版 Rev. *I です。英語版の改訂内容: Updated for ModusToolbox™ support for some of the PSoC™ 4 devices. Corrected the links in PSOC™リソース . PSoC™ 4 families are recategorized in Local sales office locations PSoC™ 4 feature set Added new section My first PSoC™ 4 design using ModusToolbox™ Removed sections : Convert Project to Bootloadable for CY8CKIT-049, Bootload Your CY8CKIT-049, and More PSoC™ 4 Code Examples
*J	2021-12-23	これは英語版 001-79953 Rev. *U を翻訳した日本語版 Rev. *J です。英語版の改訂内容: Updated to Infineon template Updated Figure 2, Figure 3, Figure 4, and Figure 5 to the latest MTB 2.3 version Updated document with PSoC™ 4100S Max device Changed the default device to PSoC™ 4100S Max and updated 図 8 Updated Modify the design section
英語版(*V)	-	この版は英語版のみです。英語版の改訂内容: Updated document with PSoC™ 4000T device Updated with the latest release version of ModusToolbox™ (version-3.0)
*K	2023-06-19	これは英語版 001-79953 Rev. *W を翻訳した日本語版 Rev. *K です。英語版の改訂内容: Updated Folder structure with the latest release version of ModusToolbox™(VERSION-3.0)
英語版(*X)	-	この版は英語版のみです。英語版の改訂内容: Updated the pin changes for PSoC™ 4000T device Updated content with the latest release of ModusToolbox™ 3.1
英語版(*Y)	-	この版は英語版のみです。英語版の改訂内容: Fixed broken links

改訂履歴

版数	日付	変更内容
英語版(*Z)	-	この版は英語版のみです。英語版の改訂内容: Updated content with the latest release of ModusToolbox™ 3.2
*L	2024-11-12	これは英語版 001-79953 Rev. AA を翻訳した日本語版 Rev. *L です。英語版の改訂内容: Updated document with PSoC™ 4100T Plus device Changed the default device to CY8CPROTO-040T kit

商標

商標

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

Trademarks

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

Edition 2024-11-12

Published by
Infineon Technologies AG
81726 Munich, Germany

© 2024 Infineon Technologies AG
All Rights Reserved.

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

Email: erratum@infineon.com

Document reference
IFX-wxy1688450359599

重要事項

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

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

警告事項

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

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