AIROC™ CYW20819/20/35 Bluetooth®および Bluetooth® LE 入門

本書について

適用範囲と目的

AN225684 は、AIROC™ CYW20819, CYW20820, および CYW20835 Bluetooth® MCU (以下、本書では CYW208xx と表記) について説明します。これらは、Bluetooth® 5.2 に準拠し、Arm® Cortex®-M4 CPU ベースの超低消費電力 MCU です。CYW20819/20 では Classic Bluetooth®と Bluetooth® Low Energy の両方に対応し、CYW20835 では Bluetooth® Low Energy に対応します。このアプリケーションノートは、CYW208xx デバイスの使用を開始する際に役立ちます。デバイスを使い始めるにあたり、デバイスアーキテクチャ, 開発キット, およびソフトウェア開発ツールの概要を説明します。また Eclipse IDE for ModusToolbox™と Bluetooth®ソフトウェア開発キット (SDK)を使用して、シンプルな Bluetooth® Low Energy アプリケーションを作成する方法を説明します。また、CYW208xx の学習を促進させるために、オンラインで利用可能なその他のリソースも案内します。

Note: AIROC™CYW20819, CYW20820, および CYW20835 は、本書では CYW208xx と表記しています。

目次

本書について 1
目次 1
1 はじめに 3
2 CYW20819, CYW20820 の概要 4
2.1 デバイスの機能 4
3 CYW20835 の概要 6
3.1 CYW20835 デバイスの機能 6
3.2 対象アプリケーション 7
4 開発エコシステム 8
4.1 ソフトウェアエコシステム – ModusToolbox™ 8
4.1.1 ModusToolbox™を使い始める 8
4.1.2 Eclipse IDE for ModusToolbox™ 8
4.1.3 CYW208xx の AIROC™ Bluetooth® SDK 9
4.1.4 コンフィギュレーター 10
4.2 開発キット 11
4.2.1 CYW920819M2EVB-01 11
4.2.2 CYW920820M2EVB-01 12
4.2.3 CYW920835M2EVB-01 13
5 初めての CYW208xx Bluetooth® Low Energy アプリケーション 14
5.1 事前準備 14
5.1.1 サンプルコードをダウンロード 15
5.2 設計について 16
5.3 パート 1: 新しいアプリケーションを作成 17
5.3.1 新しいワークスペースを選択 17
5.3.2 新しい ModusToolbox™アプリケーションを作成 17
5.3.3 CYW20820 ベースのターゲットハードウェアを選択 18

目次

5.3.4 Bluetooth® LE Find Me サンプルコードをインポート (「Using CE directly」フローにのみ適用) 19
5.3.5 スターターアプリケーションを選択し、アプリケーションを作成 (「Working from Scratch」フローにのみ適用) 20
5.4 パート 2: 設計リソースの設定 21
5.4.1 ハードウェアリソースの構成 22
5.5 共通アプリケーション設定 28
5.5.1 BT_DEVICE_ADDRESS 28
5.5.2 UART 28
5.5.3 ENABLE_DEBUG 28
5.6 Library Manager 29
5.7 パート 3: アプリケーションコードの書き込み 30
5.8 ファームウェアの説明 30
5.8.1 Bluetooth® Low Energy GATT データベース 31
5.8.2 Bluetooth®スタック設定パラメーター 31
5.8.3 ユーザーアプリケーションコードの入力 32
5.8.4 Bluetooth®スタックイベント 32
5.8.4.1 Bluetooth®スタックマネジメントイベント 32
5.8.4.2 GATT イベント 34
5.8.5 ユーザーインターフェースロジック 35
5.9 パート 4: 設計のビルド, プログラミング, およびテスト 36
6 まとめ 39
参考資料 40
改訂履歴 41
免責事項 42

1 はじめに

Bluetooth®接続を特長とするアプリケーションは、低消費電力, アプリケーションレベルの機能, および BoM コストと基板スペースの縮小に向かっています。CYW208xx Bluetooth® MCU を使用すると、これらすべての重要な要件を満たせます。

このデバイスは、以下のコア Bluetooth®機能があります。

- Bluetooth® 5.0 に準拠
- デュアルモード Bluetooth®オペレーションをサポート
- Bluetooth® Low Energy (1 Mbps および 2 Mbps) および (EDR 2 Mbps および 3 Mbps) のデータレート

一方、CYW20835 は、シングルモードの Bluetooth®動作、Bluetooth® Low Energy（1Mbps および 2 Mbps) をサポートしています。CYW208xx デバイスは、Bluetooth® Mesh 1.0 仕様を完全に実装しています。

CYW208xx は、強力な Arm® Cortex®-M4 CPU と、外部のオンボードセンサーとのインターフェースを支援する ADC, SPI, UART, および I^2C などの周辺ブロックのホストを持つことにより、付加価値アプリケーション機能にも対応します。これらの周辺ブロック, オンチップフラッシュメモリ, および統合された降圧レギュレータと LDO レギュレータの存在により、BoM コストの削減とより少ない PCB フットプリントが可能になります。インフィニオンの高度な CMOS 製造プロセスとさまざまなシステム電力モードのサポートにより、CYW208xx を使用した電池式の低電力アプリケーションを設計できます。

このデバイスは、オーディオ (ソース) , センサー (医療, 家庭, セキュリティ) , HID, リモートコントロール機能, およびその他の IoT アプリケーションのホストでの使用を目的としています。

このアプリケーションノートは、以下をカバーし、CYW208xx デバイスを使い始めるのに役立ちます。

- **CYW20819 および CYW20820 デバイスの機能とアーキテクチャの概要**
- **CYW20835 デバイスの機能とアーキテクチャの概要**
- ソフトウェア開発プラットフォーム, ハードウェア評価プラットフォーム, サンプルコード, アプリケーションノート, およびその他の関連技術ドキュメントを含む、CYW208xx デバイスの**開発エコシステム**サポートの概要。
- Eclipse IDE for ModusToolbox™で CYW20819 デバイスを使用して**アプリケーションを開発する**方法に関するチュートリアル。シンプルな Bluetooth® Low Energy ベースのアプリケーションを最初から作成する段階的なプロセスを実行します。

このアプリケーションノートでは、Bluetooth® Low Energy または従来の Bluetooth®プロトコルの詳細については説明しません。読者はこれらのプロトコルの基本にすでに精通していることを前提としています。

2 CYW20819, CYW20820 の概要

CYW20819 は、Bluetooth® Low Energy およびクラシック Bluetooth® (BR, EDR) の両方をサポートする 2.4 GHz トランシーバーを内蔵した Bluetooth® 5.2 準拠のワイヤレス MCU です。**Figure 1** に、デバイスの高レベルのアーキテクチャブロック図を示します。



Figure 1 CYW20819 ブロック図

2.1 デバイスの機能

- Bluetooth®サブシステム
  - Bluetooth®コア仕様バージョン 5.2 に準拠
  - BR, EDR 2 Mbps および 3 Mbps, eSCO, Bluetooth® LE, および LE 2 Mbps のサポートを含みます。
  - 最大+5 dBm (BR) のプログラム可能な TX 電力
  - 優れた受信感度 (Bluetooth® LE 1 Mbps で-95 dBm)
- マイクロコントローラ
  - 最大速度 96 MHz の強力な Arm® Cortex®-M4 コア
  - 256 KB のオンチップフラッシュ
  - 176 KB のオンチップ RAM
  - ROM (1 MB) に組み込まれた Bluetooth®スタック, ペリフェラルドライバー, およびセキュリティ機能により、アプリケーションはオンチップフラッシュを効率的に使用
  - AES-128 および True Random Number Generator (TRNG)
  - ECDSA 署名検証を含む ROM のセキュリティ機能
  - Over-the-air (OTA) ファームウェアのアップデート

CYW20819, CYW20820 の概要

- 有効なペリフェラル
  - 最大 22 の GPIO
  - I^2C, I^2S, UART, PCM インターフェース
  - 2 つの Quad-SPI インターフェース
  - 最大 28 のアナログチャネルを備えた補助 ADC
  - プログラム可能なキースキャン 20 x 8 マトリックス
  - 3 軸直交信号デコーダー
  - 汎用タイマーと PWM
  - Real-time clock (RTC) と watchdog timers (WDT)
- 電源管理とクロック
  - On-chip power-on reset (POR), Integrated Buck (DC-DC), および LDO レギュレータ
  - On-chip software-controlled power management unit (PMU)
  - 電力消費を最適化する複数のシステム電力モード
  - デバイス動作用の 24 MHz 外部水晶発振器 (±20 ppm 精度)
  - オプションの外部 32 kHz 水晶発振器サポートを備えたオンチップ 32 kHz 低電力発振器 (LPO)
- Wi-Fi との共存
  - インフィニオン Wi-Fi パーツで使用する Global Coexistence Interface (GCI)
  - SECI 互換 Wi-Fi パーツで使用する Serial Enhanced Coexistence Interface (SECI)
- CYW20820 はパワーアンプを内蔵しており、Tx 出力は CYW20819 が最大+4 dBm であるのに対し、最大+10 dBm まで達成可能です。

電気的仕様を含むデバイスの詳細については、**CYW20819** および **CYW20820** のデータシートを参照してください。

3 CYW20835 の概要

CYW20835 は、Bluetooth® Low Energy をサポートする 2.4GHz トランシーバーを内蔵したスタンドアロンベースバンドプロセッサを搭載し、12 dBM の出力電力で優れたレンジを実現するパワーアンプ (iPA) を内蔵した Bluetooth® 5.2 準拠の MCU です。**Figure 2** は、本デバイスのハイレベルなアーキテクチャ ブロック図です。



Figure 2 CYW20835 ブロック図

3.1 CYW20835 デバイスの機能

- Bluetooth® サブシステム
  - LE 2 Mbps をサポートする Bluetooth® コア仕様バージョン 5.2 に準拠
  - Adaptive Frequency Hopping (AFH) をサポート
  - 最大 12 dBm のプログラム可能な Tx 電力
  - Rx 感度 -94.5 dBm (Bluetooth® LE)
- マイクロコントローラ
  - Floating-Point Unit (FPU) を備えた 96 MHz Arm® Cortex®-M4 マイクロコントローラ
  - Serial Wire Debug (SWD) をサポート
  - Bluetooth® スタックおよびアプリケーションの実行
- 有効なペリフェラル
  - 6x 16 ビットの PWM
  - 24 の GPIO
  - アナログおよびデジタル マイクロフォン インターフェース

  - I2C, I2S, UART, SPI, および PCM インターフェース
  - 最大 8x20 のプログラム可能な key-scanning マトリクス インターフェース
- Wi-Fi との共存
  - グローバルな共存インターフェースをサポートし、一部のインフィニオン Wi-Fi デバイスとの共存を容易に実現
  - 電気的仕様を含むデバイスの詳細については、デバイス データシートを参照してください。

3.2 対象アプリケーション

- ウェアラブルとフィットネスバンド
- 音源アプリケーション
- Bluetooth® Low Energy Mesh Home および産業オートメーション
- 血圧モニターおよびその他の医療アプリケーション
- 近接センサー
- キーフォブ
- サーモスタットと温度計
- 玩具
- リモコン

4 開発エコシステム

4.1 ソフトウェアエコシステム – ModusToolbox™

ModusToolbox™ ソフトウェアは、多くのタイプのデバイスと環境をサポートします。実用的な観点から、ModusToolbox™ ソフトウェアは、インストール リソース, サンプルコード, BSP, ライブラリなど、さまざまな方法で提供します。利用者は必要なリソースのみを使用します。アプリケーションを作成するときは、これらのリソースを使用し、Hardware Abstraction Layer (HAL) や Peripheral Driver Library (PDL) を介してハードウェアと対話します。

Figure 3 に、ModusToolbox™ソフトウェアに含まれるツールの概要を示します。

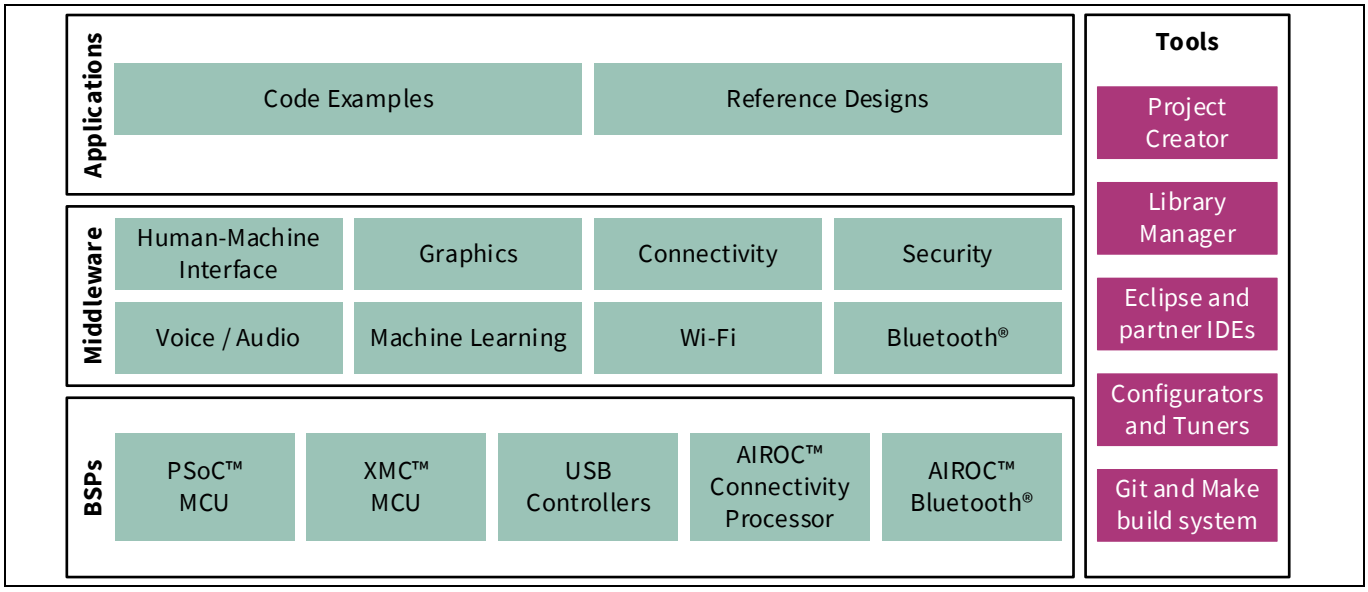

Figure 3 ModusToolbox™ソフトウェアの概要

4.1.1 ModusToolbox™を使い始める

ModusToolbox™のホームページにアクセスして、ModusToolbox™の最新バージョンをダウンロードしてインストールしてください。ModusToolbox™ソフトウェアのインストールについては、**ModusToolbox™** ホームページの *Documentation* タブにある *ModusToolbox™インストールガイド*を参照してください。インストール後、ModusToolbox™を起動し、次の項目に移動してください。

- **ユーザーガイド: Documentation** > **ModusToolbox™2.4 Documentation** > **ModusToolbox™ user guide** の詳細なユーザーガイド。
- これらのドキュメントは、**ModusToolbox™**ホームページの **Documentation** タブからも利用できます。

4.1.2 Eclipse IDE for ModusToolbox™

Eclipse IDE for ModusToolbox™は、Eclipse IDE の「Oxygen」バージョンに基づいています。Eclipse C/C++ 開発ツール (CDT) プラグインを含む、いくつかのプラグインを使用します。**Eclipse サバイバルガイド** は、Eclipse IDE for ModusToolbox™の使用方法に関するヒントを提供します。

IDE には、Eclipse 標準のメニューとツールバーに加えて、**Figure 4** に示すように、プロジェクトエクスプローラー, コードエディター, コンソールなどのさまざまなペインが含まれます。標準の Eclipse IDE との 1 つの違いは、「ModusToolbox™パースペクティブ」です。このパースペクティブは、「Quick Panel」, 「News View」を提供し、プロジェクトエクスプローラーにタブを追加します。最終的にデバイスにプログラムする最上位のエンティティは、IDE ではアプリケーションと呼ばれます。アプリケーシ

開発エコシステム

ョンは、1 つ以上の Eclipse プロジェクトで構成されています。IDE はプロジェクト間のすべての依存関係を自動的に処理します。また、ソフトウェア開発キット (SDK) によって提供されるさまざまなツールを起動するためのフックも提供します。

Eclipse IDE for ModusToolbox™を使用すると、次のことができます。

1. キットまたはデバイスでフィルター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するか、サンプルコードのコレクションをオンラインで閲覧してください。
2. ワークスペースでハードウェアシステムデザインを構築するためのデバイスリソースを構成してください。
3. ソフトウェアコンポーネントまたはミドルウェアを追加してください。
4. アプリケーションファームウェアを開発してください。

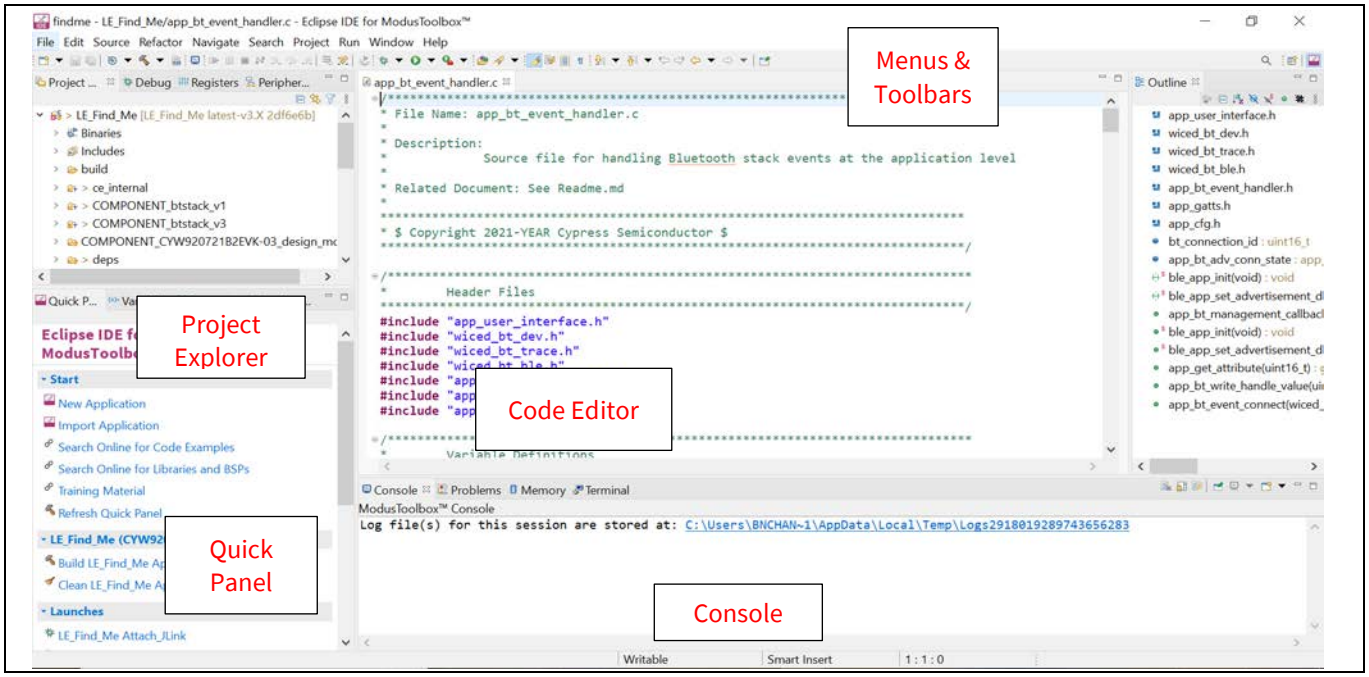

Figure 4 ModusToolbox™ IDE の概要

4.1.3 CYW208xx の AIROC™ Bluetooth® SDK

SDK は ModusToolbox™ソフトウェアの中心的なコアを提供します。SDK を使用すると、デバイスリソースの複雑な情報のすべてを理解しなくても、サポートされているデバイスのファームウェアを簡単に開発できます。SDK には、構成ツール, ドライバー, ライブラリ, ミドルウェア,さまざまなユーティリティ, Makefile, およびスクリプトが含まれています。BTSDK により、ModusToolbox™ 内での CYW208xx アプリケーション開発が可能になります。これらの BTSDK 機能は以下のとおりです。

- デバイス ROM に含まれるデュアルモード Bluetooth®スタック (BR/EDR および Bluetooth® Low Energy)
- 組込み Bluetooth®アプリケーション開発用の Bluetooth®スタックおよびプロファイルレベルの API (同じデバイス上のホストとコントローラー)
- WICED HCI プロトコルにより、より強力なプロセッサとのインターフェースを必要とするアプリケーションの開発を簡素化
- SPI, UART, ADC などのオンチップ周辺ブロックにアクセスするための API とドライバー

開発エコシステム

- サポートされる Bluetooth®プロトコルには、Generic Access Profile (GAP) , Generic Attribute Profile (GATT) , Security Manager Protocol (SMP) , Radio Frequency Communication Protocol (RFCOMM) , Service Discovery Protocol (SDP) , Bluetooth® Low Energy Mesh Protocol が含まれます。
- Bluetooth® Low Energy および BR/EDR プロファイル API, ライブラリ, サンプルアプリケーション
- Over-the-air (OTA) アップグレードのサポート
- **AIROC™ Bluetooth® SDK:** IDE で、**AIROC™ Bluetooth® SDK documentation** > **CYW208xx**> **WICED CYW208xx Documentation** を選択してください。この HTML 形式ガイドは、CYW20820 デバイスに関連する SDK API に関する情報を提供します。**Figure 5** に示すように、API 関数は論理的にコンポーネントにグループ化されており、簡単に移動して参照できます。

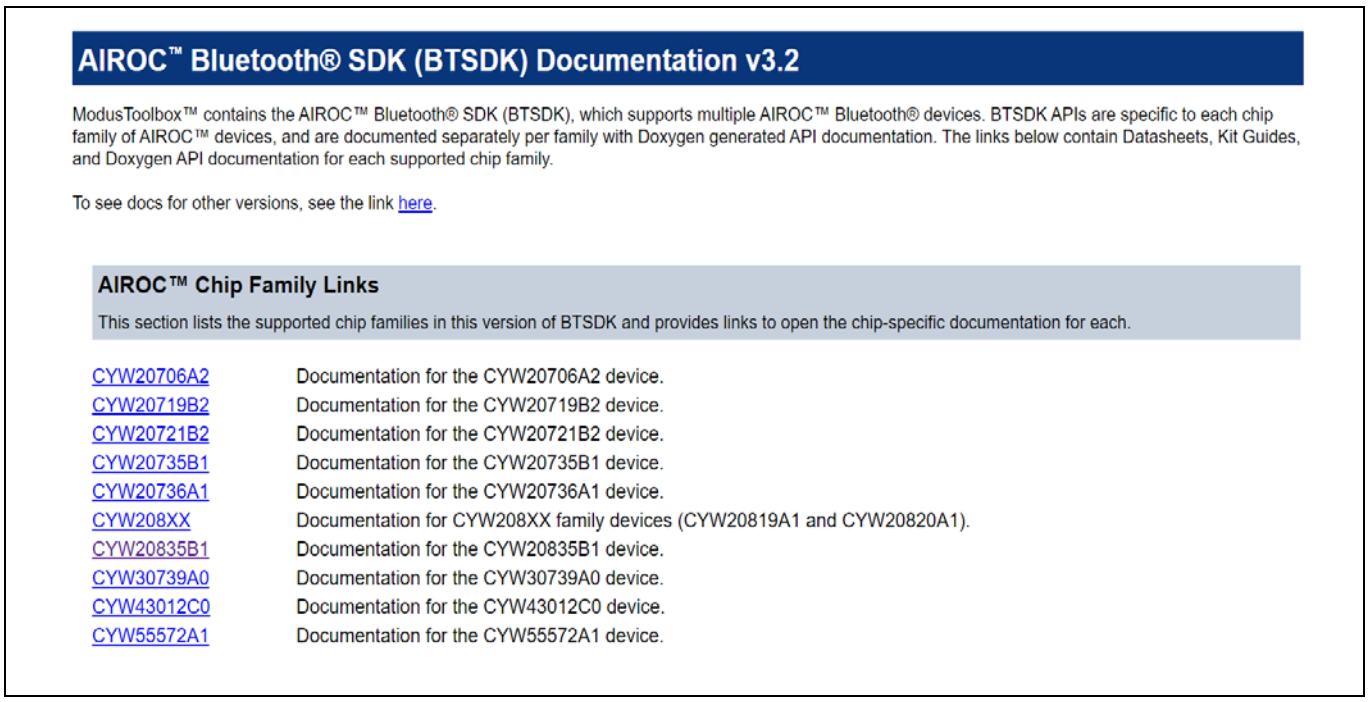

Figure 5 Bluetooth® SDK API リファレンス

SDK でサポートされている詳細な機能の一覧については、**AIROC™ API reference** を参照してください。

4.1.4 コンフィギュレーター

ModusToolbox™ソフトウェアは、ハードウェアリソース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。CYW208xx に適用できるコンフィギュレーターを以下に示します。

- **Device Configurator** は、アプリケーションで使用されるペリフェラルとピンを有効化/設定するために使用されます。
- **Bluetooth® Configurator** は、アプリケーションの Bluetooth® Low Energy GATT データベースを生成します。

次のセクションでは、Bluetooth® Low Energy アプリケーション作成演習の一部としてこれらのコンフィギュレーターを使用する方法を詳しく見ていきます。

4.2 開発キット

4.2.1 CYW920819M2EVB-01

Figure 6 に示す **CYW920819M2EVB-01** は、CYW20819 を使用したプロトタイピングとアプリケーション開発をサポートする開発キットです。

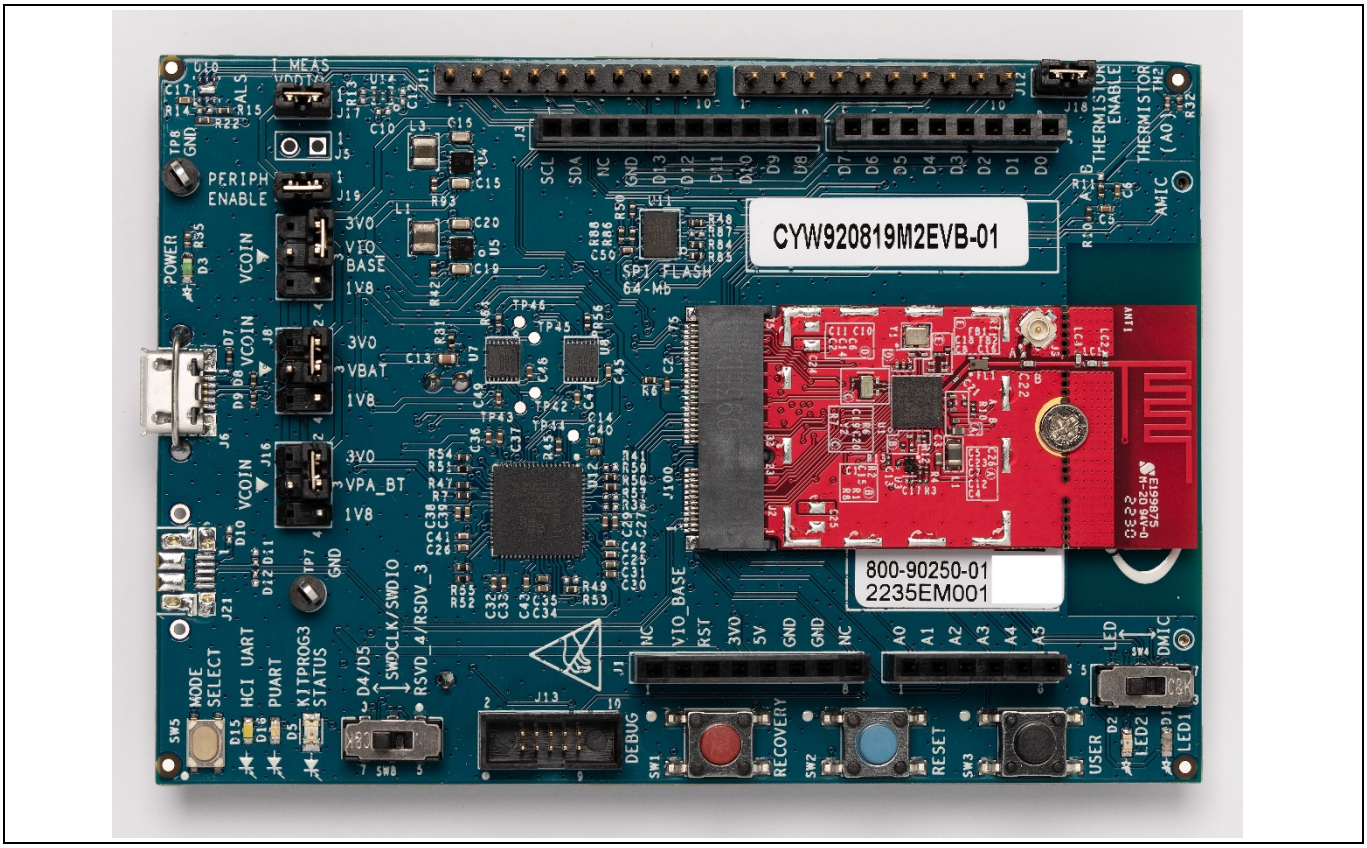

Figure 6 CYW920819M2EVB-01 評価ボード

このキットは、次の主要機能をサポートしています。

1. オンボードアンテナ付き CYW20819 キャリアボード
2. Arduino シールドと互換性のある拡張ヘッダー
3. CYW20819 デバイスの 1.8 V および 3.3 V 動作のサポート
4. リカバリーボタン (オレンジ) , リセットボタン (青) , ユーザーボタン (黒) , 2 つのユーザーLED, アナログサーミスタ, およびモーションセンサー
5. USB-to-UART デバイスを介して CYW20819 デバイスの 2 つのシリアルインターフェース (HCI UART およびペリフェラル UART (PUART)) へのアクセスを提供するオンボードマイクロ USB コネクタ。通常、HCI UART はアプリケーションコードのダウンロードを使用し、PUART は Bluetooth®スタックとアプリケーショントレースメッセージの出力、または UART を介した外部デバイスとのインターフェースに使用されます。

ユーザーガイドや回路図など、キットの詳細については、**CYW920819M2EVB-01** Web ページを参照してください。

4.2.2 CYW920820M2EVB-01

Figure 7 に示す **CYW920820M2EVB-01** は、CYW20820 を使用したプロトタイピングとアプリケーション開発をサポートする開発キットです。

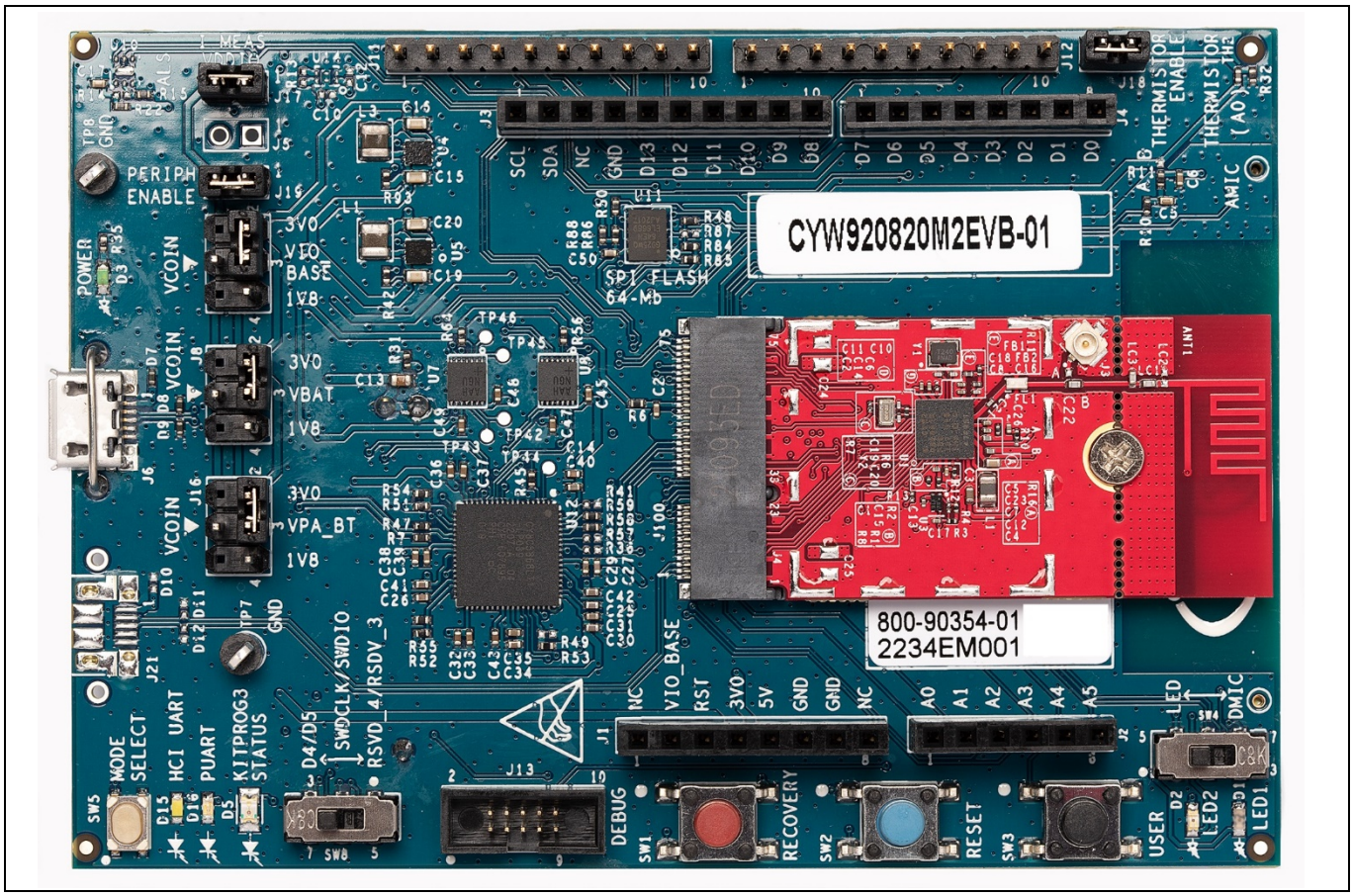

Figure 7 CYW920820M2EVB-01 評価ボード

このキットは、次の主要機能をサポートしています。

1. オンボードアンテナ付き CYW20820 キャリアボード
2. Arduino シールドと互換性のある拡張ヘッダー
3. CYW20820 デバイスの 1.8 V および 3.0 V 動作のサポート
4. リカバリーボタン (オレンジ) , リセットボタン (青) , ユーザーボタン (黒) , 2 つのユーザーLED, アナログサーミスタ, およびモーションセンサー
5. USB-to-UART デバイスを介して CYW20820 デバイスの 2 つのシリアルインターフェース (HCI UART およびペリフェラル UART (PUART)) へのアクセスを提供するオンボードマイクロ USB コネクタ。通常、HCI UART はアプリケーションコードのダウンロードを使用し、PUART は Bluetooth®スタックとアプリケーショントレースメッセージの出力、または UART を介した外部デバイスとのインターフェースに使用されます。

ユーザーガイドや回路図など、キットの詳細については、**CYW920820M2EVB-01** Web ページを参照してください。

4.2.3 CYW920835M2EVB-01

Figure 8 に示す **CYW920835M2EVB-01** は、CYW20835 を使用したプロトタイピングとアプリケーション開発をサポートする開発キットです。

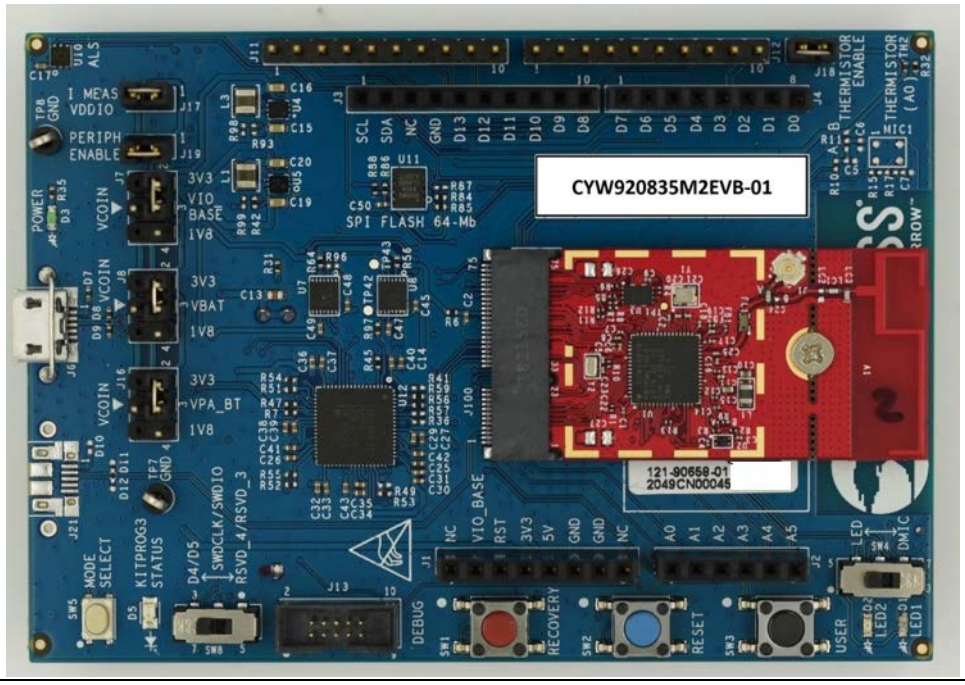

Figure 8 CYW920835M2EVB-01 評価ボード

このキットは、次の主要機能をサポートしています。

1. 60-QFN CYW20835 Bluetooth® Low Energy 5.2 準拠 MCU
2. オンボードセンサー: 環境光センサー, サーミスタ, アナログマイク用のフットプリント, および インフィニオン デジタルマイク用のフットプリント
3. ユーザースイッチと LED
4. 電源, プログラミング, および USB-UART ブリッジ用の USB コネクタ

ユーザーガイドや回路図など、キットの詳細については、**CYW920835M2EVB-01** Web ページを参照してください。

5 初めての CYW208xx Bluetooth® Low Energy アプリケーション

ここでは、Eclipse IDE for ModusToolbox™を使用して CYW20820 デバイス用の簡単な Bluetooth® Low Energy ベースのアプリケーションを構築するための手順を説明します。**Find Me Profile (FMP)** と呼ばれる Bluetooth® SIG 定義の標準プロファイルが設計に実装されています。このセクションで説明する手順は次のとおりです。

- **パート 1：新しいアプリケーションを作成**
- **パート 2：設計リソースの設定**
- **パート 3：アプリケーションコードの記述**
- **パート 4: 設計のビルド, プログラミング, およびテスト**

これらの手順では、特定のサンプルコード (この場合は `BLE_FindMe`) を使用する必要があります。ただし、サンプルコード (CE) を使用する範囲は、これらの手順を実行するために選択したパスによって異なります。このドキュメントでは、サンプルコード (CE) とアプリケーションという用語は同じ意味で使用されています。サンプルコード (CE) は、特定の目的または機能を提供する既存の ModusToolbox™アプリケーションです。

学習したい内容に応じて、これらの手順を通じて 2 つのパスを定義しました。

パス	「Using CE directly」パス (既存のサンプルコード (CE) を直接評価)	「Working from Scratch」パス (既存のサンプルコード (CE) を参照としてのみ使用)
対象	ツールまたはデバイスの初心者で、すべてがどのように機能するかをすばやく確認したい人。	ModusToolbox™で CYW208xx ベースの Bluetooth®アプリケーションを開発する方法をハンズオンで学びたい人。

各パスで行う必要があることは、インストラクションの各パートの始めに明確に定義されています。

ゼロから始めて、このアプリケーションノートのすべての指示に従う場合は、指示に従うときにサンプルコードを参照として使用します。ゼロから作業することは、設計プロセスを学ぶのに役立ち、そして時間がかかります。あるいは、既存のサンプルコードを直接評価して、短時間で CYW208xx アプリケーション開発フローに慣れることができます。

どちらの場合も、まず **事前準備** と **設計について** を読んでから始めるのが役立ちます。

5.1 事前準備

この演習では、次のアイテムを用意してください。

- PC にインストールされている **ModusToolbox™** 2.4 以降のバージョン
- **CYW920819M2EVB-01**/**CYW920820M2EVB-01**/**CYW920835M2EVB-01** 評価ボード
- LightBlue **iOS**/**Android** アプリ、または Immediate Alert Service (IAS) をサポートするその他の Android または iOS アプリ。

LightBlue アプリをダウンロードするために、携帯電話で以下の QR コードをスキャンしてください。

IOS Android

5.1.1 サンプルコードをダウンロード

サンプルコードは、GitHub で使用可能です。**GitHub** から PC 上の場所に `mtb-example-btsdk-ble-findme` サンプルコードリポジトリを複製またはダウンロードしてください。これを行うには、3 つの方法があります。

1. **Figure 9** に示すように、**mtb-example-btsdk-ble-findme** に移動し、Clone or download オプションを使用して、サンプルコードリポジトリを zip ファイルとしてダウンロードしてください。ダウンロードしたファイルをハードドライブ上の場所に解凍してください。このアプローチは、Git バージョン管理システムに慣れていない場合に役立ちます。

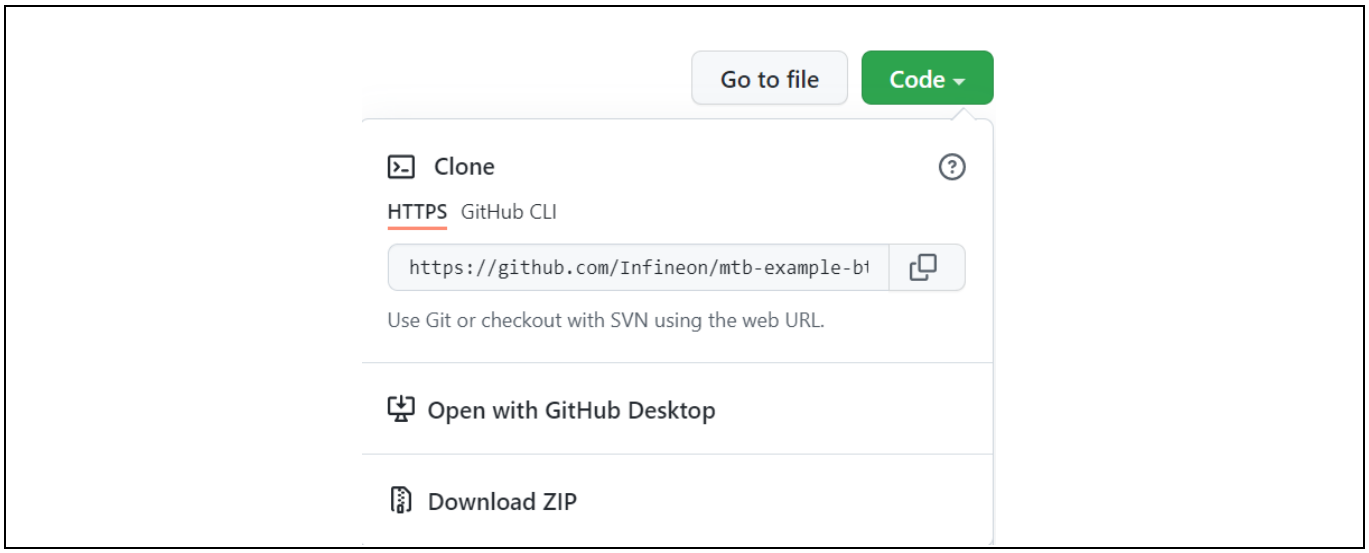

Figure 9 GitHub からサンプルコードリポジトリをダウンロード

2. Eclipse IDE for ModusToolbox™で **File > Import > Git > Projects from Git** を選択し、ウィザードの指示に従ってください。これは標準の Eclipse 機能です。
3. `git clone` コマンドまたは GitHub デスクトップクライアントを使用してください。

PC にリポジトリを作成すると、Bluetooth® LE Find Me プロファイルのサンプルコードは、`mtb-example-btsdk-ble-findme` のリポジトリフォルダー内にあります。このサンプルコードは、次のセクションで従うことを選択した開発フローに応じて、参照として使用するか、直接使用します。

Note: *Eclipse IDE for ModusToolbox™には、**Figure 10** に示すように、**Quick Panel** のオンラインサンプルコードにアクセスするためのリンクも用意されています。**Search Online for Code Examples** リンクをクリックしてください。これにより、GitHub リポジトリへの Web ページが開き、適切なリポジトリを選択してダウンロードしてください。次に、上記の手順を使用して、必要なリポジトリをダウンロードできます。*

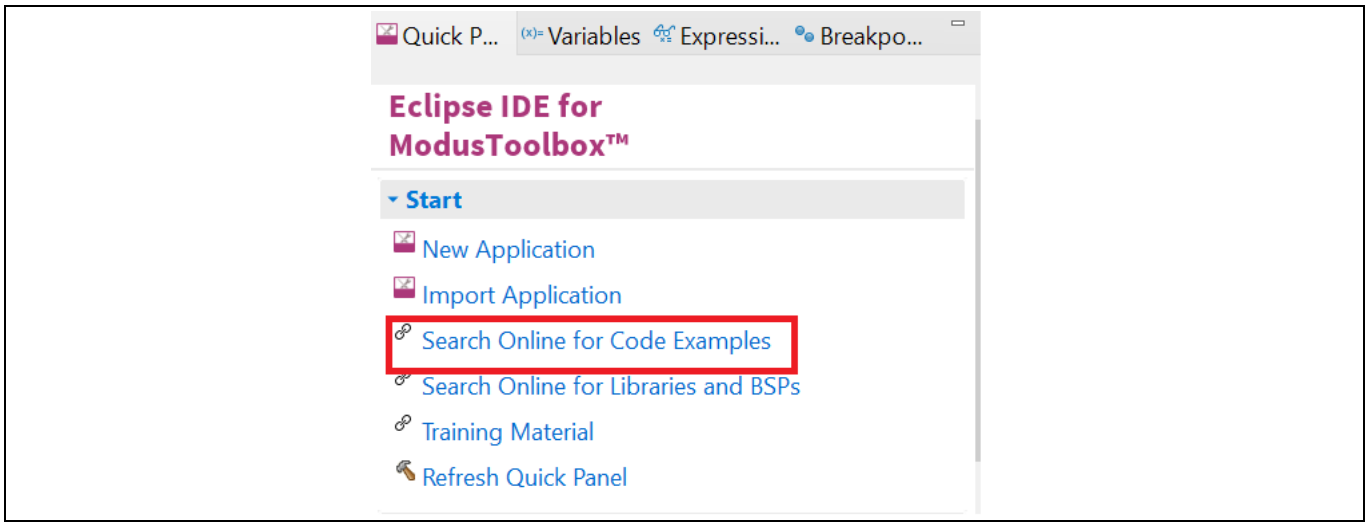

Figure 10　サンプルコードをオンラインで検索するためのクイック起動オプション

5.2　設計について

この設計は、Immediate Alert Service (**IAS**) で構成される Bluetooth® Low Energy **Find Me Profile (FMP)** を実装しています。FMP と IAS は、それぞれ **Bluetooth® SIG** で定義されている Bluetooth® Low Energy 標準プロファイルとサービスです。

この設計では、CYW920820M2EVB-01 キットの 2 つの LED (赤色 LED, 黄色 LED) を使用しています。赤色 LED (LED2) は IAS アラートレベル (アラートなし (LED オフ), 軽度のアラート (LED 点滅), 高アラート (LED オン))を表示します。黄色の LED (LED1) は、周辺機器 (CYW20820) が通知 (LED 点滅), 接続 (LED オン), または切断 (LED オフ) しているかを示します。さらに、デバッグ UART インターフェースは、Bluetooth®スタックとアプリケーショントレースメッセージの送信に使用されます。

iOS/Android モバイルデバイスまたは PC は、周辺機器に接続する Bluetooth® Low Energy セントラルデバイスとして機能できます。

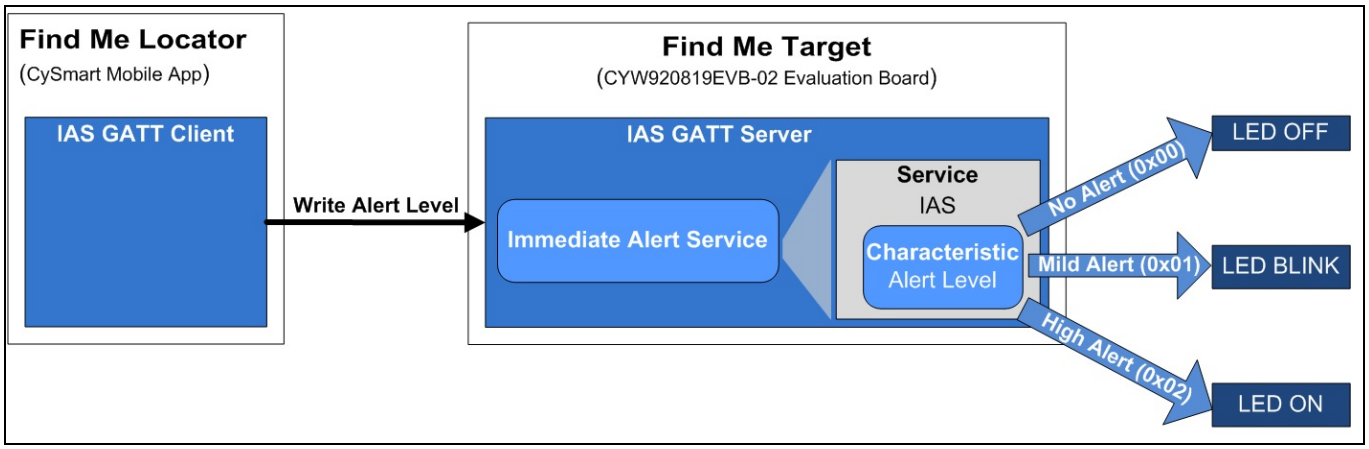

Figure 11　CYW208xx を使用した Find Me Profile (FMP) アプリケーション

5.3 パート 1: 新しいアプリケーションを作成

このパートでは、新しい ModusToolbox™アプリケーションを作成するプロセスをステップごとに説明します。このセクションの手順を実行する前に、サンプルコードをそのままインポートして実行するか、アプリケーションを最初から作成する方法を学ぶかを決定します。選択に応じて、実行する必要がある手順は次のとおりです。

パス	「Using CE directly」パス (既存のサンプルコード (CE) を直接評価)	「Working from Scratch」パス (既存のサンプルコード (CE) を参照としてのみ使用)
アクション	5.3.1,5.3.2, 5.3.3, 5.3.4 に従ってください。5.3.5 は無視してください。	5.3.1, 5.3.2, 5.3.3, 5.3.5 に従ってください。5.3.4 は無視してください。

ModusToolbox™を起動して開始してください。

5.3.1 新しいワークスペースを選択

起動時に、ModusToolbox™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を選択するダイアログを表示します。ワークスペースディレクトリは、ワークスペース環境設定と、デバイス構成やアプリケーションソースコードなどの開発成果物を格納するために使用されます。

Figure 12 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と完全なパスを入力すると、ModusToolbox™がディレクトリを作成します。

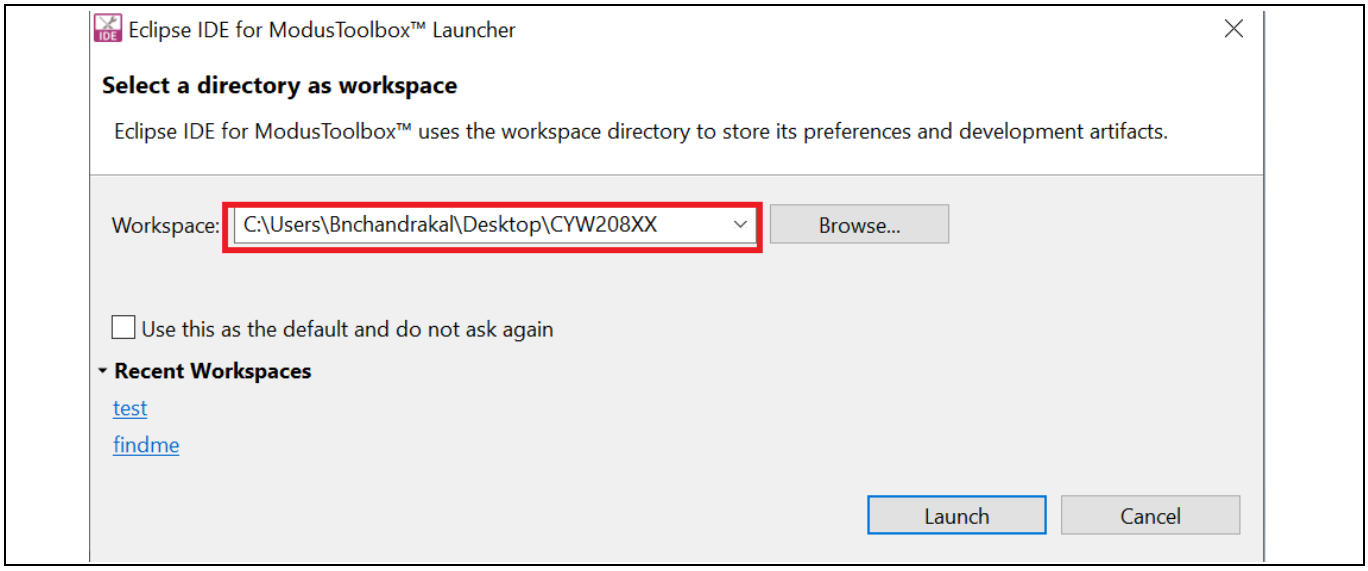

Figure 12 ワークスペースとしてディレクトリを選択

5.3.2 新しい ModusToolbox™アプリケーションを作成

クイックパネルの Start グループで **New Application** をクリックしてください。または、**File > New > ModusToolbox™ Application** を選択することもできます (Figure 13)。

Eclipse IDE for ModusToolbox™アプリケーションウィンドウが表示されます。

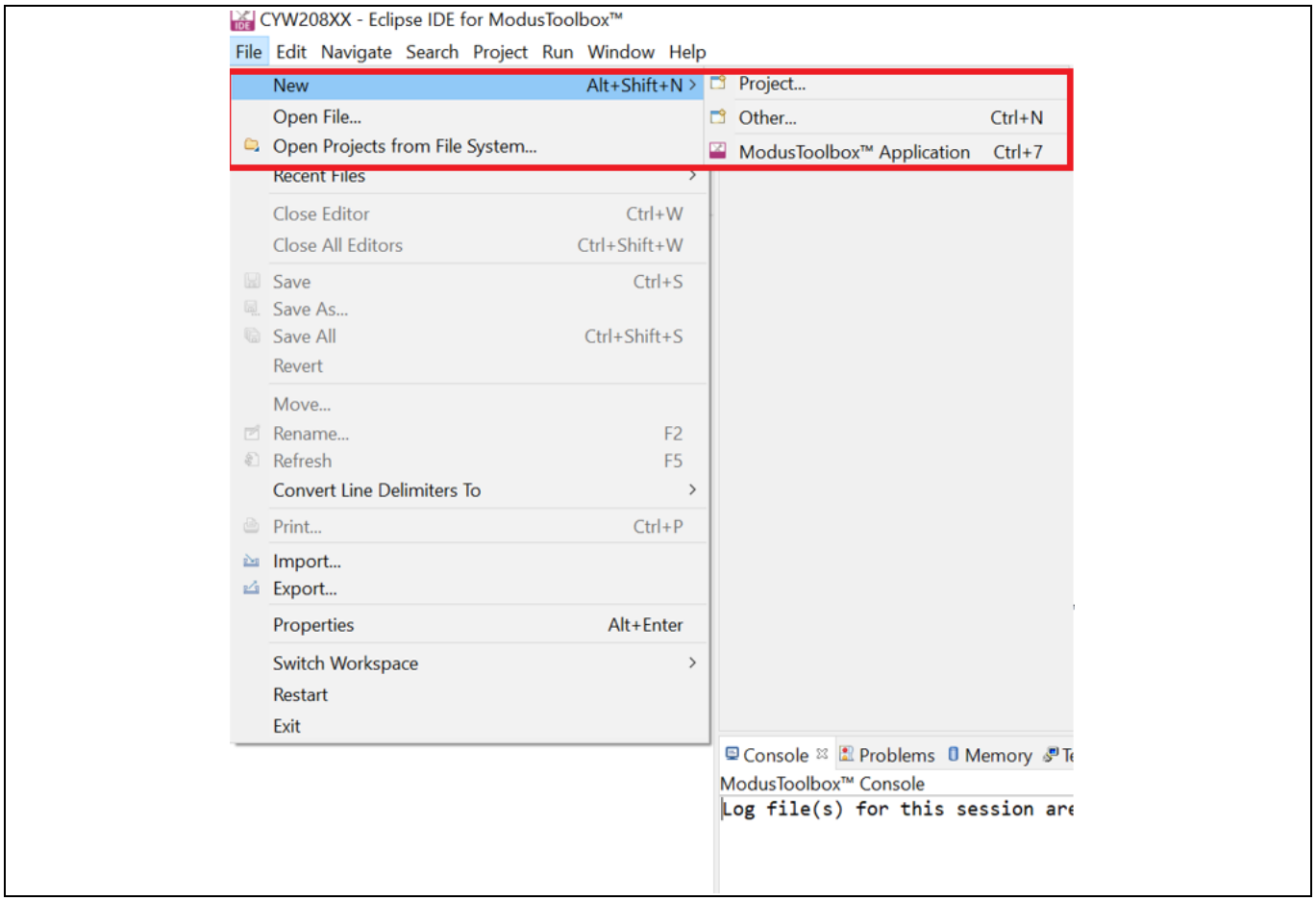

Figure 13　新しい ModusToolbox™アプリケーションを作成する

5.3.3　CYW20820 ベースのターゲットハードウェアを選択

ModusToolbox™は、アプリケーション開発を開始するためのインフィニオンキットのリストを表示します。今回は、CYW20820 デバイスを利用した CYW920820M2EVB-01 評価ボード上でのアプリケーション開発を行いたいと思います。**CYW920820M2EVB-01** を選択し、**Next** をクリックしてください (**Figure 14**)。

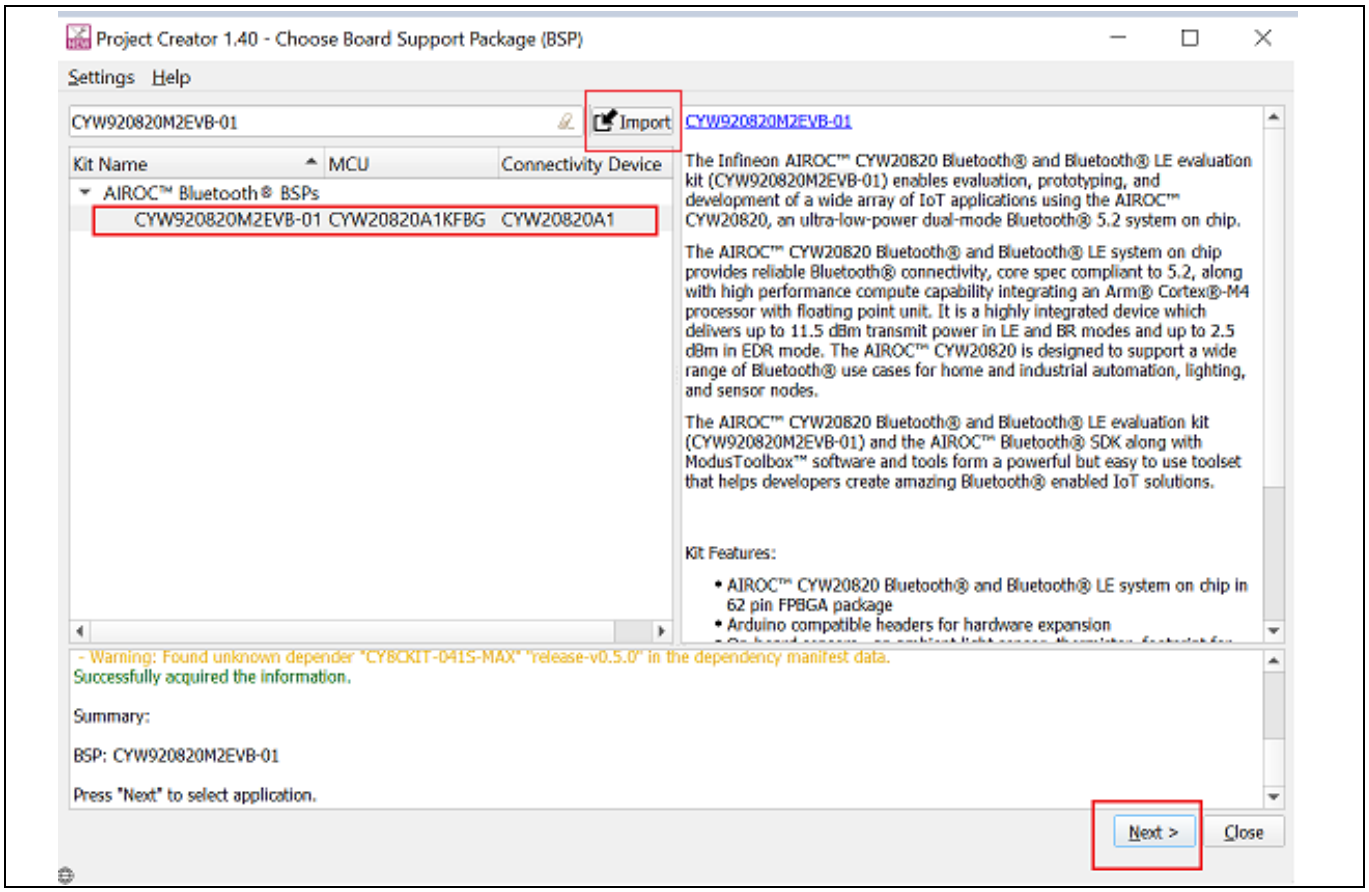

Figure 14　ターゲットハードウェアを選択

Note: ***カスタム BSP*** *オプション: CYW20820 デバイスに基づくカスタムボードがある場合は、* ***Figure 14*** *に示すようにカスタム BSP を作成し、Import ボタンでインポートしてください。カスタム BSP を作成する手順については、ドキュメント* ***Creating Custom BSP*** *を参照してください。このアプリケーションノートでは「カスタム BSP」開発フローを使用していませんが、以下の手順は「カスタム BSP」開発フローのガイドとして使用できます。*

5.3.4　Bluetooth® LE Find Me サンプルコードをインポート (「Using CE directly」フローにのみ適用)

Import オプションは、リポジトリからダウンロードした、または同僚から受け取ったサンプルコードを Eclipse IDE for ModusToolbox™にインポートするのに役立ちます。この機能を使用して、Using CE directly フローの Bluetooth® LE Find Me サンプルコードをインポートしてください。**Figure 15** に示す **Bluetooth® LE Find Me サンプルコードのインポート** ダイアログで、**Import** をクリックし、前にダウンロードした CE リポジトリ (`mtb-example-btsdk-ble-findme`) の Bluetooth® LE Find Me サンプルコードの場所に移動してください。サンプルコードフォルダーを選択してください。Bluetooth® LE Find Me サンプルコードの名前と説明が **Starter Application** ダイアログに表示されます。必要に応じてアプリケーション名を変更してください。**Next** をクリックし、その後 **Finish** をクリックして、アプリケーションの作成プロセスを完了してください。

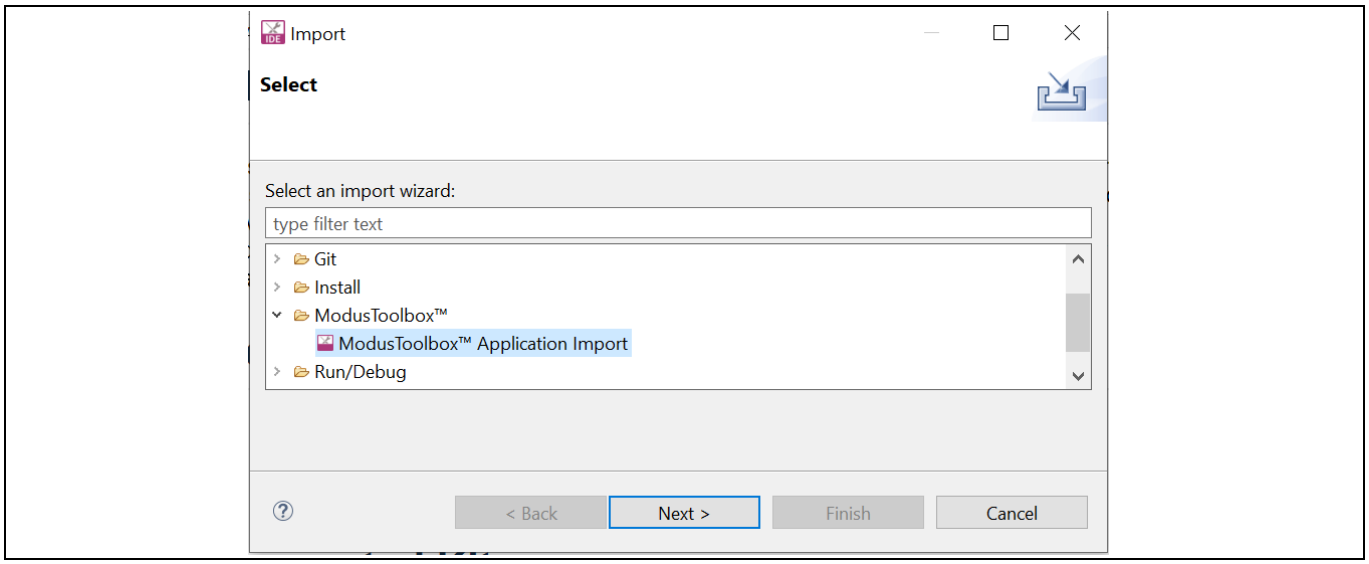

Figure 15 Bluetooth® LE Find Me サンプルコードのインポート

CYW20820 用の新しい ModusToolbox™アプリケーションが正常にインポートされました。

5.3.5 スターターアプリケーションを選択し、アプリケーションを作成 (「Working from Scratch」フローにのみ適用)

Working from Scratch 開発フローの開始点として、既存のテンプレートアプリケーションを使用します。**Figure 16** に示す **Starter Application** ダイアログで、**Empty_BTSDK_App** を選択してください。**Name** フィールドに、アプリケーションの名前を入力して、**Next** をクリックしてください。アプリケーションの要約ダイアログが表示されます。**Finish** をクリックして、ModusToolbox™にアプリケーションを作成してください。

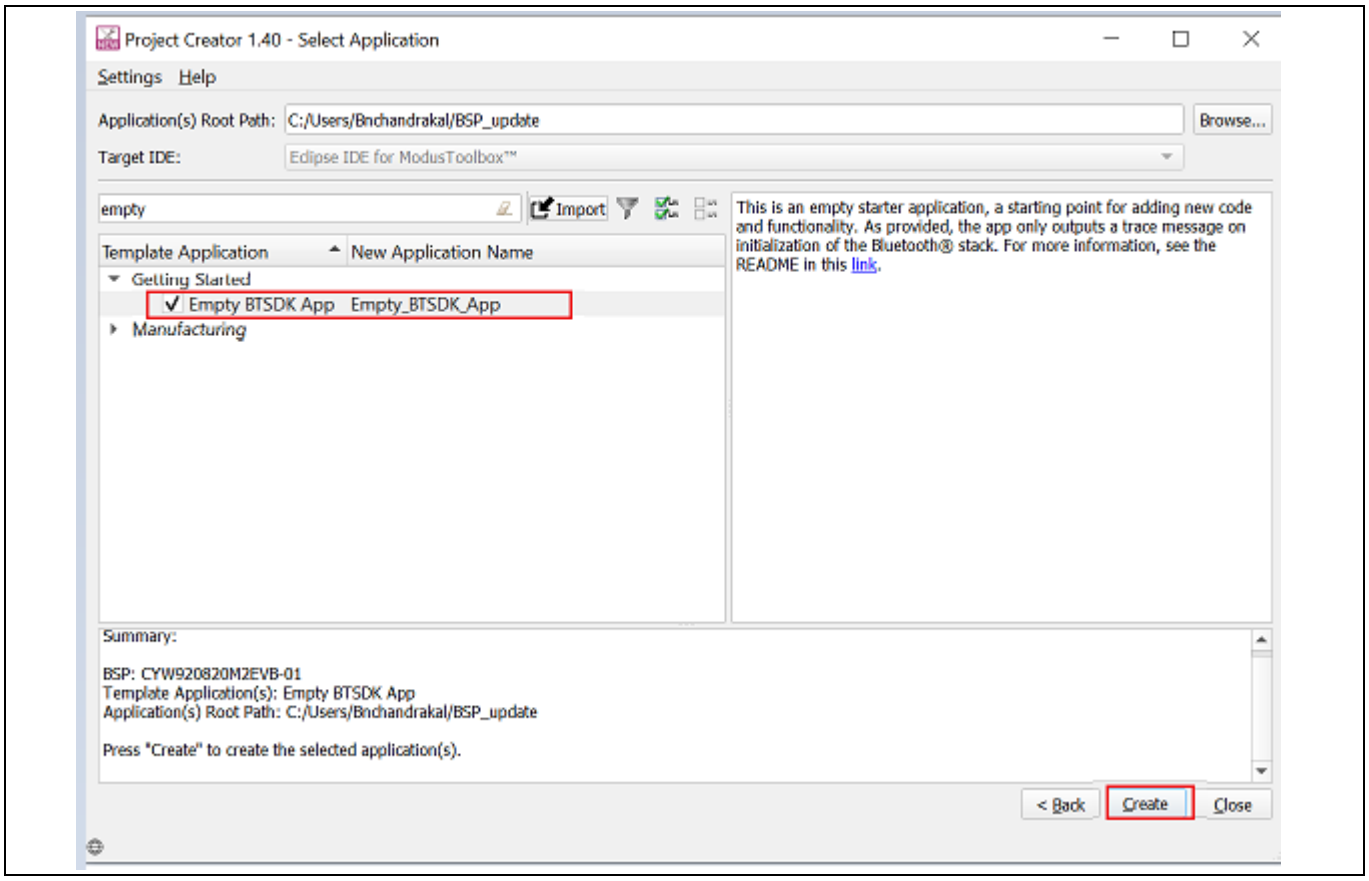

Figure 16　スターターアプリケーションウィンドウ

CYW20820 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。

5.4　パート 2: 設計リソースの設定

このステップでは、アプリケーションの設計リソースを設定し、設定コードを生成します。

パス	「Using CE directly」パス (既存のサンプルコード (CE) を直接評価)	「Working from Scratch」パス (既存のサンプルコード (CE) を参照としてのみ使用)
アクション	すべてのステップを読んで理解してください。CE はすでにすべてのリソース構成を完了しているため、このセクションの手順を実行する必要はありません。	すべてのステップを行う

Empty_BTSDK_App アプリケーションテンプレートには、CYW902820M2EVB-01 キットで利用可能なすべてのリソースが事前設定されており、すぐに使用できます。これらのリソースには、ユーザーLED, プッシュボタン, 通信ペリフェラル (Bluetooth®, UART, I^2C, SPI) が含まれます。テンプレートアプリケーションには、デバイスと Bluetooth®スタックを初期化し、Peripheral UART (PUART) インターフェースにステータスメッセージを出力するデフォルトのアプリケーションプログラムコードも含まれています。

先に進む前に、ModusToolbox™プロジェクトエクスプローラーのクイックツアーが用意されています。**Figure 17** に、テンプレートアプリケーションが作成された直後の ModusToolbox™プロジェクトエクスプローラービューを示します。

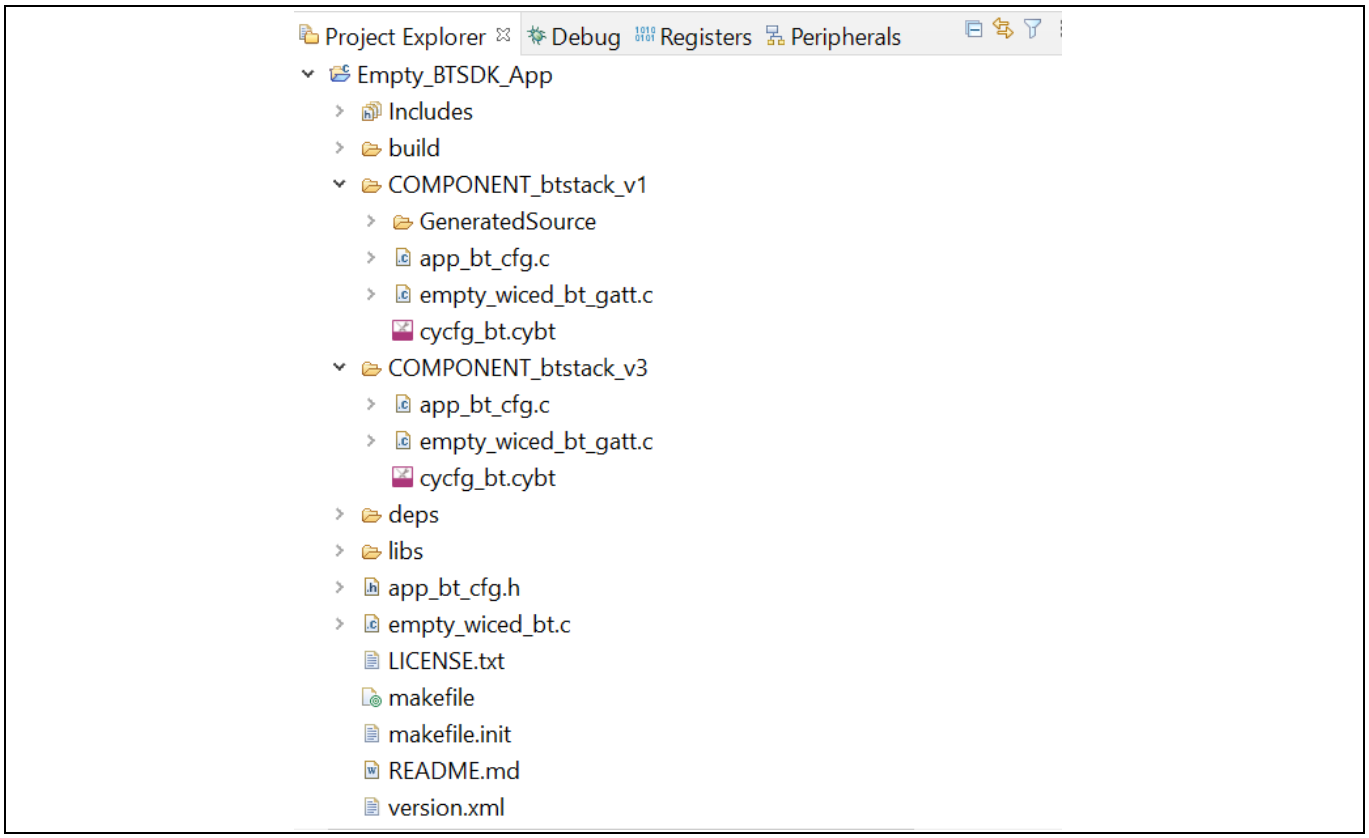

Figure 17　プロジェクトエクスプローラービュー

- *Includes* フォルダーには、Bluetooth®スタック, hardware abstraction layer (HAL), ハードウェアプラットフォームなどに関連するヘッダーファイルが含まれます。API 関数のプロトタイプ, 定数, およびデフォルトは、これらのヘッダーファイルに含まれます。記述するアプリケーションコードは、必要に応じてこれらの API 関数と定数を使用します。
- `GeneratedSource` フォルダーには、このアプリケーションの設計設定が含まれます。これには、アプリケーションに対して有効化/無効化されたリソースのリスト、およびさまざまなリソースの設定が含まれます。このフォルダーの内容については、アプリケーション開発フローの次のステップで詳しく説明します。
- `app_bt_cfg.c` および `app_bt_cfg.h` ファイルには、デバイス名などのランタイム Bluetooth®スタック設定パラメーター、およびアドバタイズ/接続設定が含まれます。
- `empty_wiced_bt.c` ファイルには、BTSTACK バージョン 3.0 より前のすべてのデバイス用の関数が含まれます。
- *readme.txt* ファイルには、アプリケーションの機能を説明する情報が含まれます (この場合は、`Empty_BTSDK_App`) 。
- *makefile.init* はビルドプロセスの一部として使用されます。これにより、上級ユーザーはビルドフローで新しいルールと変数を指定できます。
- `Makefile` には、基本および高度な設定パス情報が含まれます。

それでは、テンプレートアプリケーションでデザインリソースの設定方法を見ていきましょう。

5.4.1　ハードウェアリソースの構成

Empty_BTSDK_App テンプレートアプリケーションからアプリケーションをセットアップしたので、このアプリケーションに必要なハードウェアリソースを設定し、対応する設定コードを生成します。

初めての CYW208xx Bluetooth® Low Energy アプリケーション

Eclipse IDE for ModusToolbox™は、アプリケーションの設定を `design.modus` ファイルに保存します。このファイルは、設定ファームウェアを生成するグラフィカルコンフィギュレーターによって使用されます。このファームウェアは、アプリケーションの `GeneratedSource` フォルダーに格納されています。

Device Configurator は、アプリケーションで使用されるペリフェラルとピンを有効化/設定するために使用されます。Device Configurator を起動するには、`design.modus` ファイルをダブルクリックするか、**Figure 18** に示すように、クイックパネルで **Device Configurator** をクリックしてください。DeviceConfigurator で変更を行うたびに、**File > Save** をクリックして、更新された設定を保存してください。

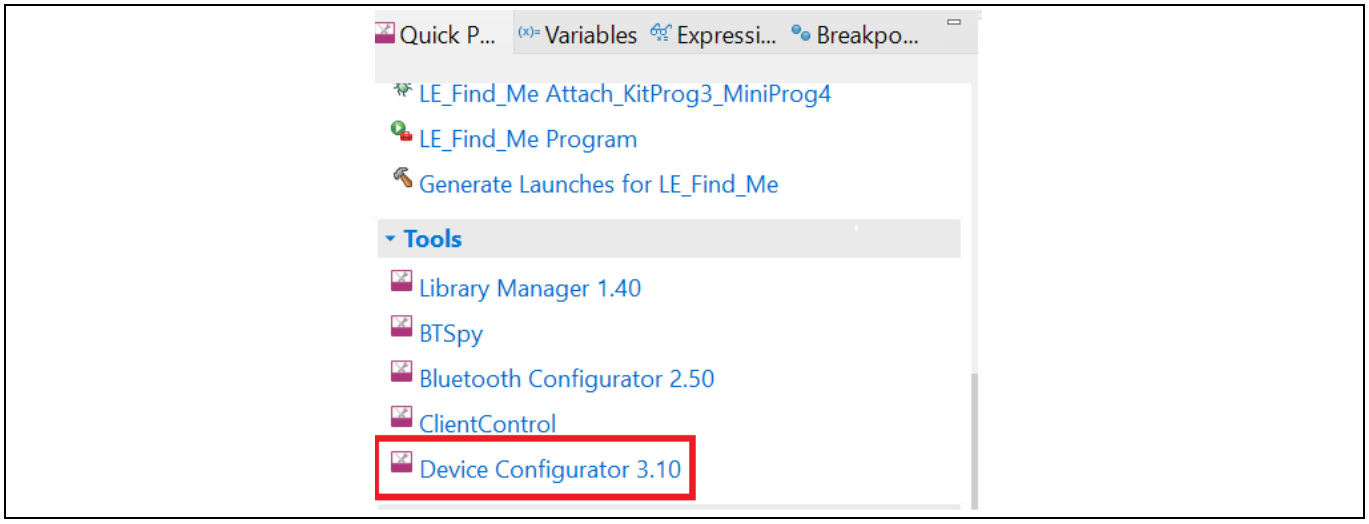

Figure 18　クイックパネルビュー

Figure 19 に、このアプリケーションの Peripherals ビューを示す Device Configurator ビューを示します。

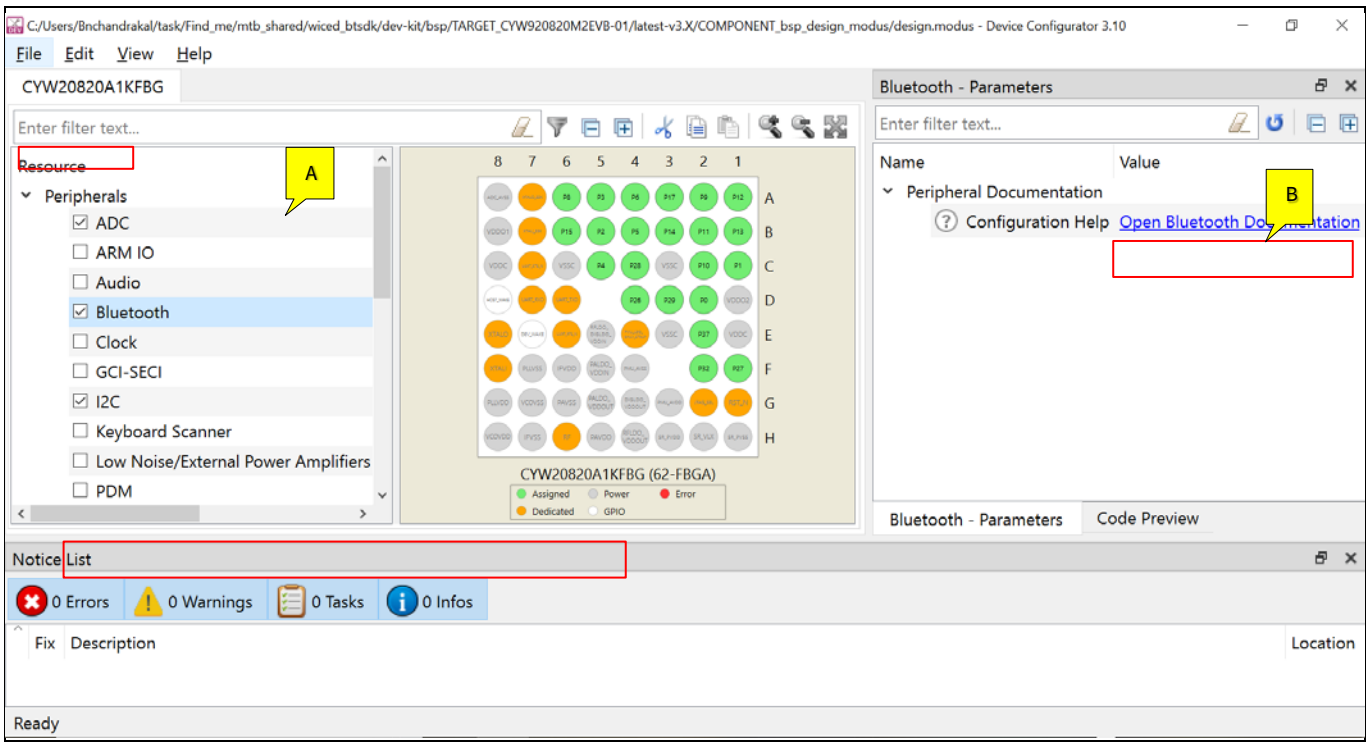

Figure 19　Device Configurator - Peripherals

- **Peripherals:** テンプレートアプリケーションでは、**Figure 19** に示すように、キットのペリフェラルがデフォルトで有効です (Bluetooth®, I²C, SPI 1, UART 2)。Find Me アプリケーションでは、Bluetooth®および PUART (UART 2) ペリフェラルのみが使用されます。ただし、テンプレートアプリケーションでのデフォルトのペリフェラルの有効化/無効化設定はそのままにしておくことができます。I²C と SPI はコードで有効化されないため、コンフィギュレーターで有効化してもかまいません。
  ペリフェラルを設定するには、**Resource** ペインでペリフェラル名をクリックしてください。**Parameters** ペインには、そのペリフェラルに提供されている設定オプションが表示されます。 コンフィギュレーターのドキュメントを開いて詳細を確認するには、Device Configurator で **Help** -> **View Help** を選択してください。
- **Bluetooth® Configurator:** Bluetooth®ペリフェラルには、Bluetooth® Configurator と呼ばれる追加のコンフィギュレーターがあり、アプリケーションの Bluetooth® Low Energy GATT データベースを生成するために使用されます。Find Me Profile アプリケーションの場合、CYW20820 デバイスの Find Me Target ロールに対応する GATT データベースを生成する必要があります。それを行う前に、IDE または PC ファイル エクスプローラーから `cycfg_bt.cybt` および `app_bt_cfg.c` ファイルを `COMPONENT_btstack_v1` フォルダーからプロジェクトの最上位フォルダーに移動します。CYW20820/CYW20819/CYW20835 デバイスは btstack v1 をサポートしているため、これらのファイルをコピーしています。完了したら、IDE または PC ファイル エクスプローラーから `COMPONENT_btstack_v1` フォルダーと `COMPONENT_btstack_v3` フォルダーを削除します。GATT データベースを設定するには、`cycfg_bt.cybt` ファイルをダブルクリックするか、**Figure 20** に示すように **Quick Panel** で **Bluetooth Configurator** を選択して、Bluetooth® Configurator を起動します。Bluetooth® Configurator のデフォルト ビューを **Figure 21** に示します。

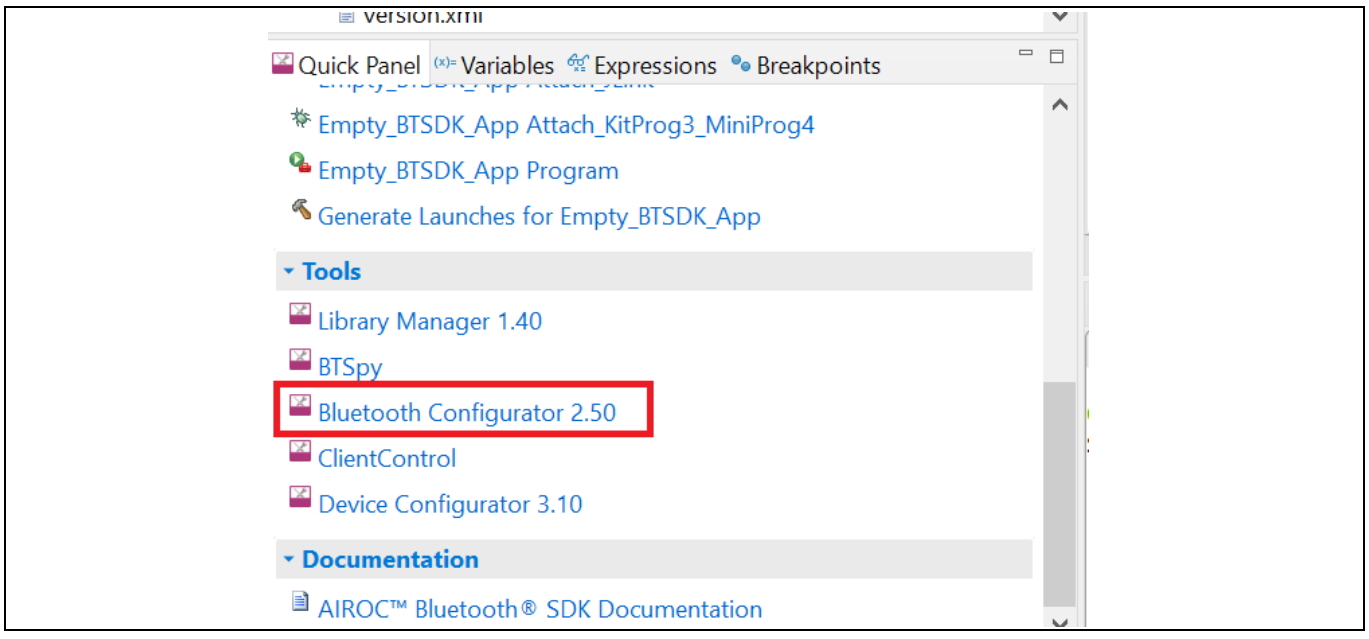

Figure 20　クイックパネルビュー - Bluetooth® Configurator

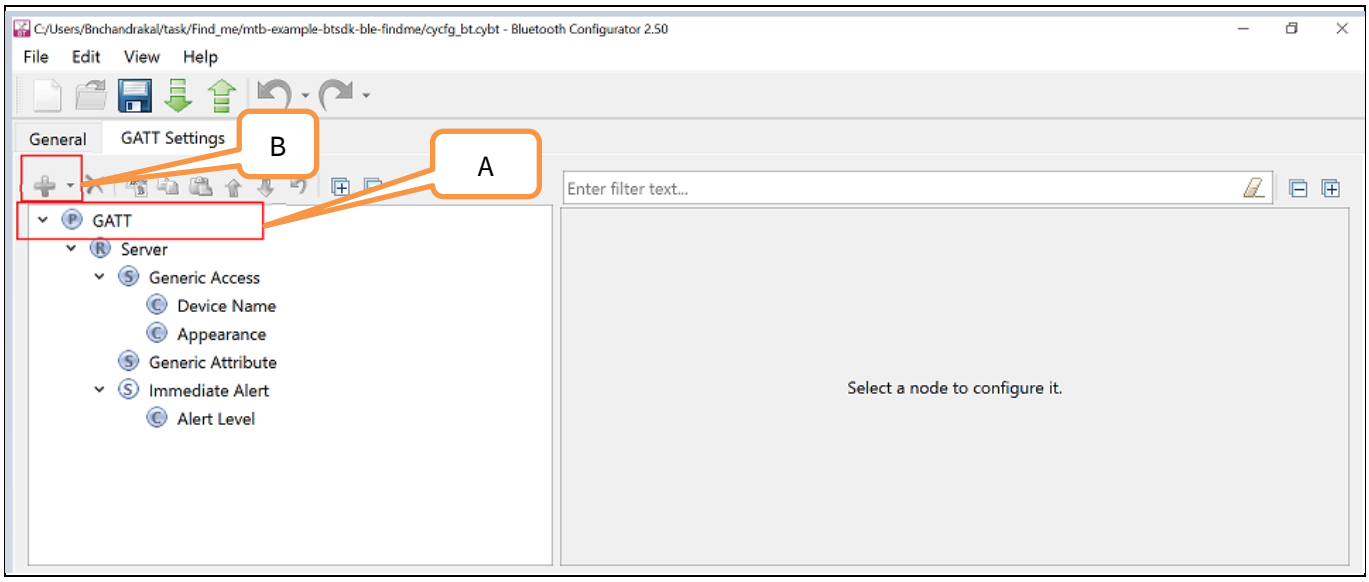

Figure 21 Bluetooth® Configurator

Find Me Target プロファイルを追加するためには、**GATT** プロファイル (**Figure 21** の A) をクリックし、次に+アイコン (**Figure 21** の B) をクリックしてください。**Figure 22** に示すように、ドロップダウンメニューから **Find Me Target (GATT Server)** プロファイルを選択してください。

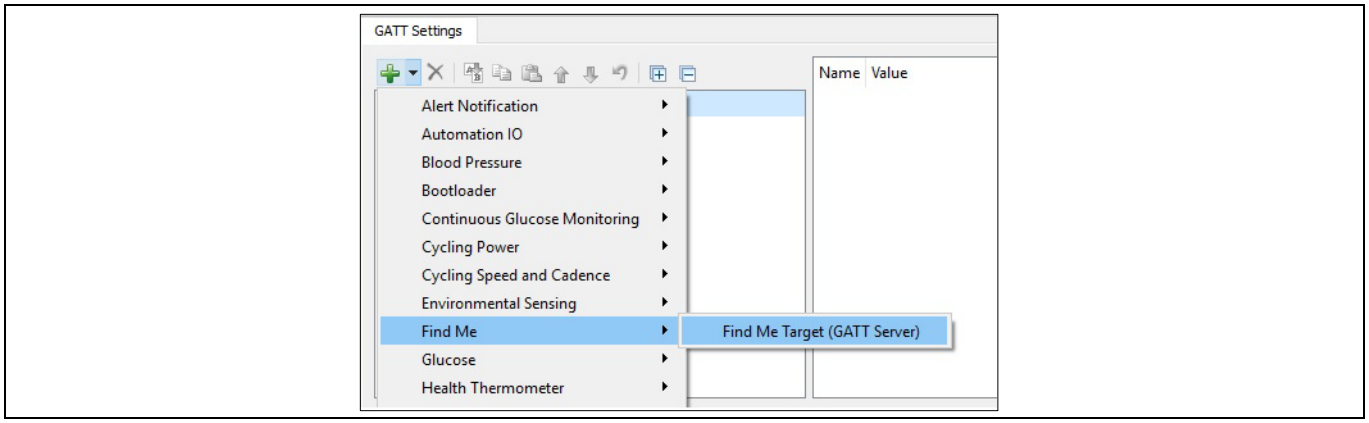

Figure 22 Find Me ターゲットプロファイルの追加

Figure 23 に、Find Me Target Server が追加された後の GATT データベースビューを示します。Find Me Target プロファイルに対応する Immediate Alert Service が **Figure 23** に追加されていることに注意してください。Configurator ウィンドウで **File** > **Save** をクリックするか、Save アイコンをクリックしてください。コンフィギュレーターは、GATT データベースを `GeneratedSource` フォルダーのソースファイル `cycfg_gatt_db.h` および `cycfg_gatt_db.h` に保存してください。同じフォルダー内の `cycfg_bt.h` ファイルには、XML 形式の Bluetooth®コンフィギュレーター設定が含まれており、コンフィギュレーターは、起動時に設定をロードするために使用します。

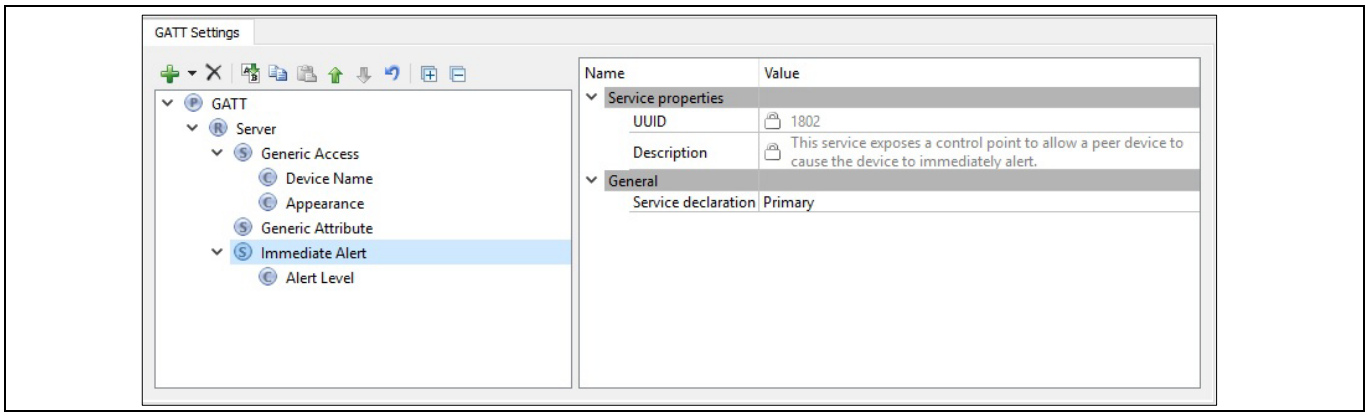

Figure 23　最終的な GATT データベースビュー

- **Pins**: アプリケーションで使用されるピンは、**Figure 24** に示すように、デバイスコンフィギュレーターの **Pins** セクションで有効になります。テンプレートアプリケーションには、LED, ボタン, I²C, SPI, PUART などのペリフェラルに使用されるピンなど、キットの機能に対応したいくつかのピンがすでに有効化され、設定されています。Find Me アプリケーションの場合、対象となるピンは 2 つの LED ピン (LED1, LED2) と PUART インターフェースピン (UART_TX, UART_RX) です (**Table 1** を参照)。これらのピンのパラメーター設定は、**Figure 25**, **Figure 26**, **Figure 27**, および **Figure 28** に示すようにデフォルト設定のままにします。SPI および I²C ピンは変更しません。これらのピンは他の目的には使用しません。そのため、使用しているボードのデフォルトのペリフェラルにピンを割り当てても問題ありません。Device Configurator で変更を行うときはいつでも、**File > Save** をクリックするか、Save アイコンをクリックして、更新された設定を保存してください。

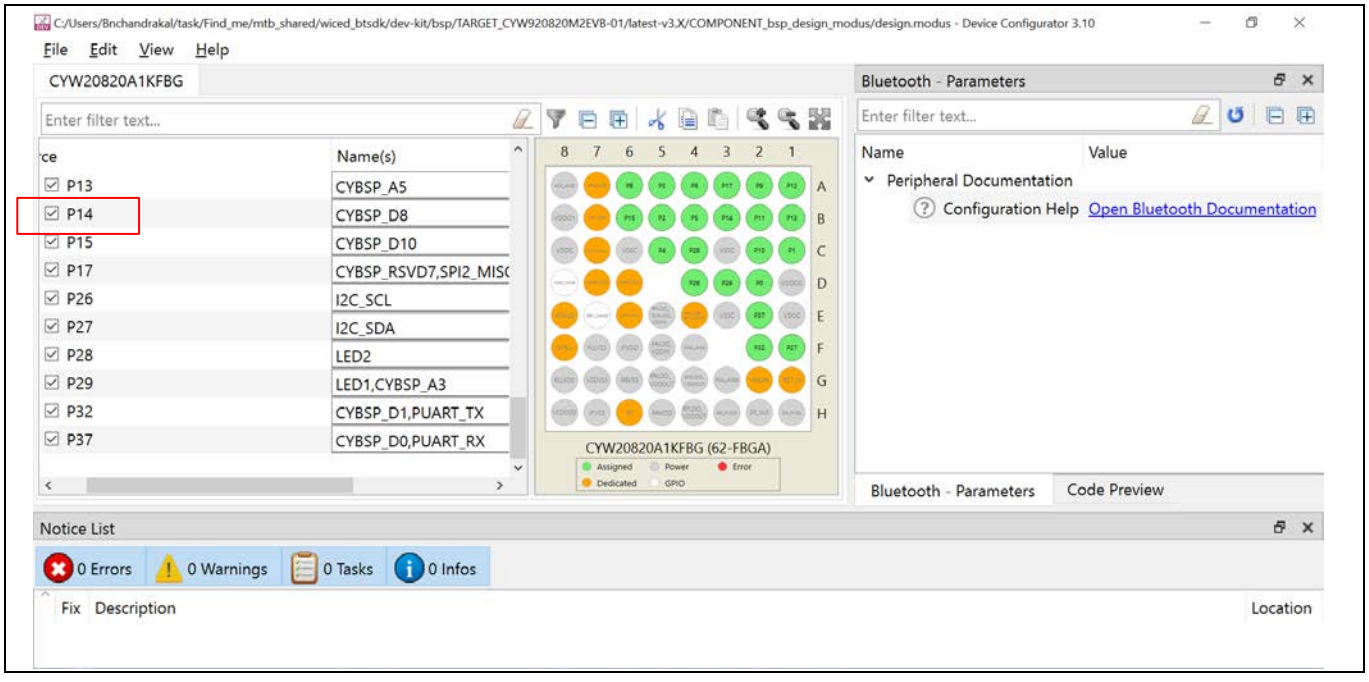

Figure 24　デバイスコンフィギュレーター-ピン

Table 1　　ピンマッピングの詳細

端子	エイリアス	用途	設定
P28	LED2	キットの赤色 LED (LED2) にマッピングされています。IAS 警告レベルを示します。	Figure 25 と Figure 26 を参照してください。図に示すように、各 LED のインデックス設定が異なることを確認してください。コンフィギュレーターはこのチェックを行いません。インデックス値も 1 から開始する必要があります (0 ではなく)。インデックス値が設定コードによってどのように使用されるかを確認するには、cycfg_pins.h を参照してください。
P29	LED1	キットの黄色の LED (LED1) にマッピングされています。Bluetooth® Low Energy 周辺機器の Advertising / Connected 状態を示します。	
P32	PUART_TX	PUART ペリフェラルの Tx ピン	Figure 27 を参照してください。
P37	PUART_RX	PUART ペリフェラルの Rx ピン	Figure 28 を参照してください。

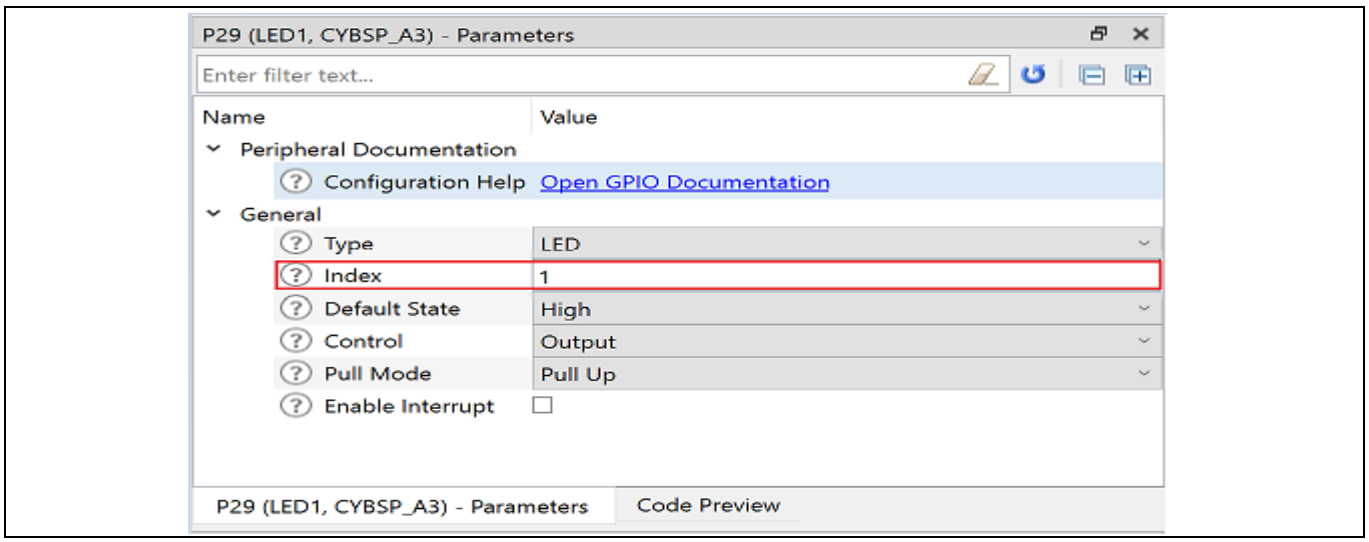

Figure 25　LED1 ピン設定

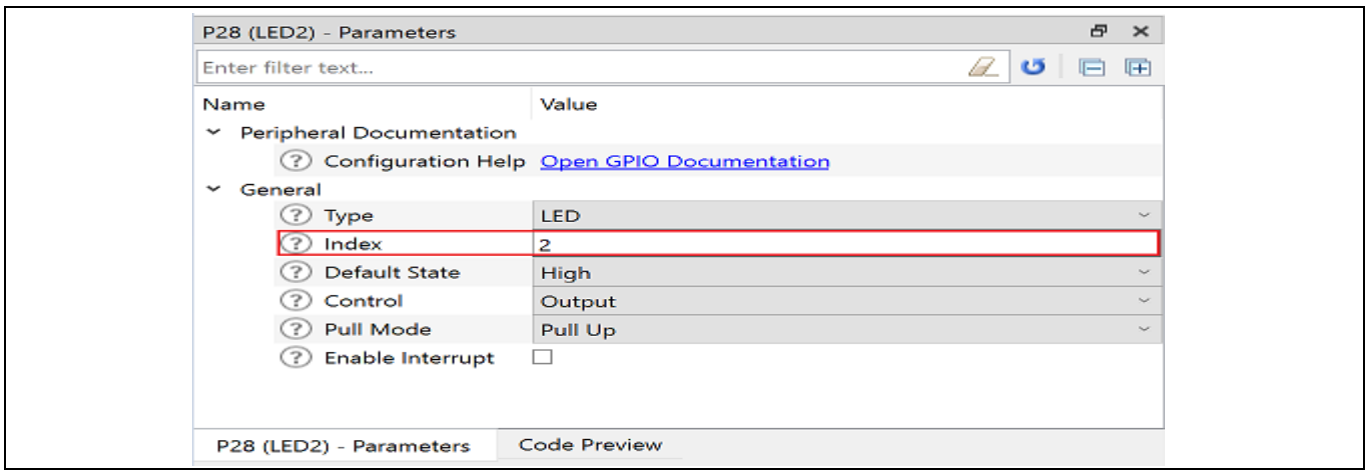

Figure 26　LED2 ピン設定

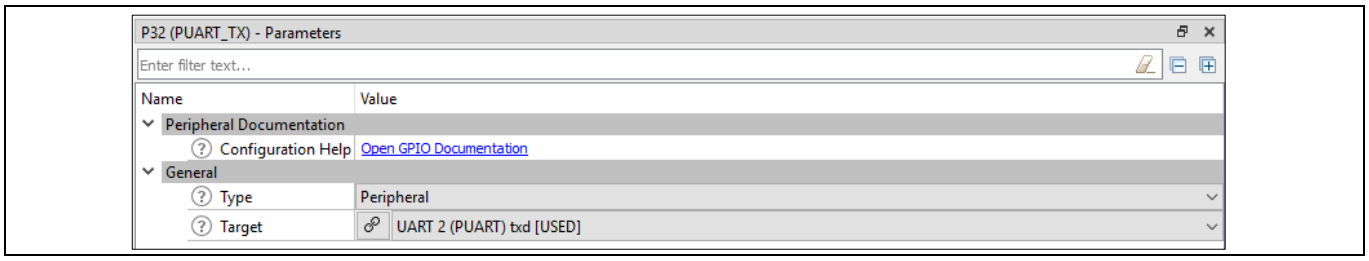

Figure 27 PUART Tx ピン設定

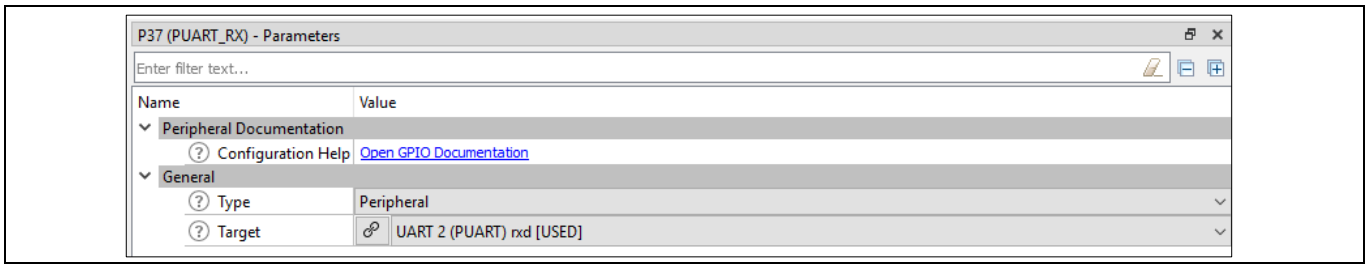

Figure 28 PUART Rx ピン設定

これでアプリケーションのリソース構成が完了し、必要な設定コードが GeneratedSource フォルダーのファイルに生成され、設定も design.modus ファイルに保存されました。

5.5 共通アプリケーション設定

以下のアプリケーション設定は、すべての BTSDK アプリケーションに共通であり、アプリケーションの makefile を介して設定するか、コマンド ラインを介して渡すことができます。

5.5.1 BT_DEVICE_ADDRESS

デバイスの BDA (Bluetooth® デバイス アドレス) を設定してください。アドレスは 6 バイトです (例: 20819A10FFEE)。デフォルトでは、SDK は 7 桁の 16 桁のデバイス ID とホスト PC の MAC アドレスの最後の 5 桁の 16 進数を組み合わせて、デバイスの BDA を設定します。

5.5.2 UART

アプリケーションのダウンロードに使用する UART ポートを設定してください。例えば、Windows では「COM6」, Linux では「/dev/ttyWICED_HCI_UART0」, macOS では「/dev/tty.usbserial-000154」です。SDK はデフォルトで、ポートを自動検出します。

5.5.3 ENABLE_DEBUG

ハードウェア デバッグの場合は、ENABLE_DEBUG=1 を設定します。この設定は、SWD の GPIO を設定します。詳細については、ドキュメント **AIROC™-Hardware-Debugging** を参照してください。

- CYW920819M2EVB-01/CYW920820M2EVB-01: SWD 信号は D4 および D5 と共有されます。回路図の SW9 を参照してください。
- CYBT-213043-MESH/CYBT-213043-EVAL/CYBT-253059-EVAL: SWD 信号は、P12=SWDCK および P13=SWDIO にルーティングされます。拡張コネクタを使用して、VDD, GND, SWDCK, および SWDIO を SWD デバッガプローブに接続します。

- CYBT-223058-EVAL/CYW920835M2EVB-01/CYBT-243053-EVAL/CYBLE-343072-EVAL-M2B/CYBLE-333074-EVAL-M2B/CYBLE-343072-MESH: SWD 信号は、P02=SWDCK および P03=SWDIO にルーティングされます。拡張コネクタを使用して、VDD, GND, SWDCK, および SWDIO を SWD デバッガ プローブに接続します。
- CYBT-263065-EVAL/CYBT-273063-EVAL: SWD 信号は、P02=SWDCK および P04=SWDIO にルーティングされます。拡張コネクタを使用して、VDD, GND, SWDCK, および SWDIO を SWD デバッガ プローブに接続します。
- SWD ハードウェアのデバッグは、以下ではサポートされていません。
  - CYW920835M2EVB-01
  - CYW920819M2EVB-01/CYW920820M2EVB-01

5.6 Library Manager

ModusToolbox™ の Bluetooth® SDK は、Bluetooth® アプリケーションを開発するためのさまざまなミドルウェア コンポーネントを選択するための「Library Manager」ダイアログを提供します。Library Manager ダイアログを起動するには、Project Explorer でトップレベルのプロジェクト名を右クリックし、**ModusToolbox™** > **Library Manager** オプションを選択してください (このオプションについては、**Figure 29** を参照してください) 。または、クイック パネルの **Select Library Manager** リンクをクリックすることもできます。

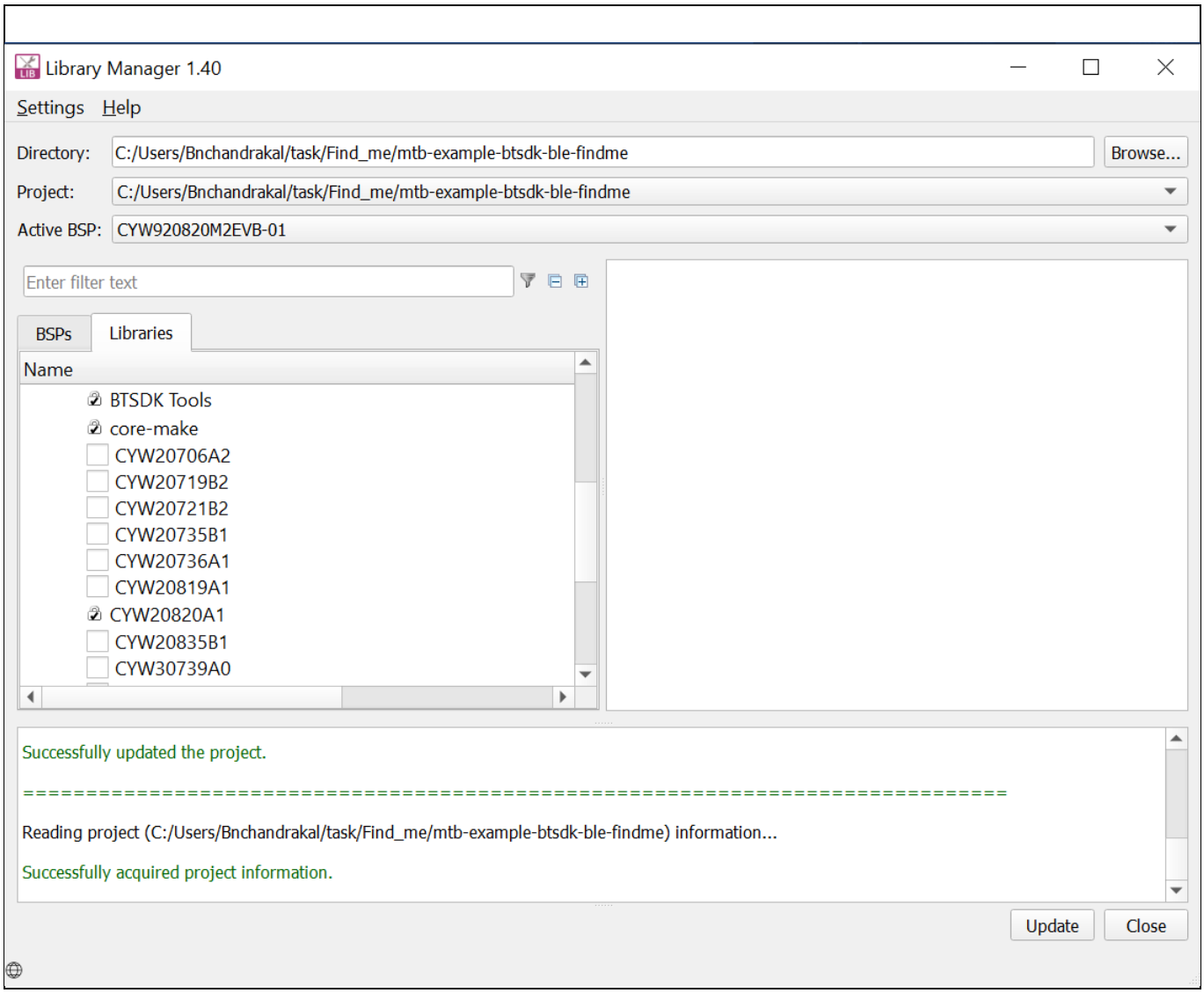

Figure 29 Library Manager

5.7 パート 3: アプリケーションコードの書き込み

開発プロセスのこの時点で、アプリケーションを作成し、ハードウェアリソースを設定し、Bluetooth® Low Energy GATT データベースを含む設定コードを生成しました。ここでは、Find Me Target 機能を実装するアプリケーションコードを調べます。

パス	「Using CE directly」パス (既存のサンプルコード (CE) を直接評価)	「Working from Scratch」パス (既存のサンプルコード (CE) を参照としてのみ使用)
アクション	手順 1 は無視してください。CE には必要なソースファイルがすべて追加されています。**ファームウェアの説明**セクションを読んで、ファームウェアの設計を理解してください。	ステップ 1 を実行してください。**ファームウェアの説明**を読み、ファームウェアの設計を理解してください。

アプリケーションコードは、以下に示す主要なタスクを実行する必要があります。

- Bluetooth®スタックを含むシステム初期化を実行してください。
- アドバタイズ, 接続, および属性の読み出し/書き込み要求などのさまざまなイベントの Bluetooth®スタックイベントハンドラー関数を実装してください。
- ユーザーインターフェースロジックを実装して、トリガーされたイベントに基づいてキットの LED 状態を更新してください。

コードを理解およびコードの再利用を可能にするために、上記の各タスクが個別のソース/ヘッダーファイルに含まれるモジュール式ファームウェア開発アプローチを使用します。

1. プロジェクトにファイルを追加する (「Working from Scratch」フローでのみ必要)

- リポジトリからダウンロードした ***mtb-example-btsdk-ble-findme*** サンプルコードを見つけてください。リポジトリのダウンロードプロセスとサンプルコードの場所については、**事前準備**セクションを参照してください。
- 以下のファイルを `mtb-example-btsdk-ble-findme` サンプルコードの最上位フォルダーから ModusToolbox™ワークスペースフォルダー内の *mtb-example-btsdk-ble-findme* フォルダーにコピーしてください。
  - `find_me.c`
  - `app_bt_event_handler.c`
  - `app_bt_event_handler.h`
  - `app_gatts.c`
  - `app_gatts.h`
  - `app_user_interface.c`
  - `app_user_interface.h`
- `empty_wiced_bt.c` ファイルを Eclipse IDE for ModusToolbox™プロジェクトエクスプローラーまたは PC ファイルエクスプローラーから削除してください。同等の `application_start()`関数が前のステップでコピーされた `find_me.c` ファイルの一部になっているため、このファイルは不要になりました。

5.8 ファームウェアの説明

ここでは、Find Me アプリケーションのアプリケーションファームウェアについて説明します。このサンプルコードのユーザーアプリケーションレベルのコードに関連する重要なソースファイルを **Table 2** に示します。

Table 2　重要なユーザーアプリケーション関連のソースファイル

ファイル名	備考
cycfg_gatt_db.c, cycfg_gatt_db.h	これらのファイルは、アプリケーションフォルダーの下の *GeneratedSource* フォルダーにあります。これらには、Bluetooth® Configurator ツールを使用して生成された GATT データベース情報が含まれます。
app_bt_cfg.c app_bt_cfg.h	これらのファイルには、デバイス名やアドバタイズ/接続設定などのランタイム Bluetooth®スタック設定パラメーターが含まれます。
Find_me.c	デバイスの起動後にユーザーアプリケーションコードを実行するためのエントリポイントである application_start()関数が含まれます。
app_bt_event_handler.c app_bt_event_handler.h	これらのファイルには、Bluetooth®スタックイベントハンドラー関数のコードが含まれます。
app_user_interface.c app_user_interface.h	これらのファイルには、アプリケーションユーザーインターフェース (この場合は LED) 機能のコードが含まれます。
app_gatts.c app_gatts.h	これらのファイルには、スタックからの GATT イベントを処理する GATT 関数のコードが含まれます。

5.8.1　Bluetooth® Low Energy GATT データベース

cycfg_gatt_db.c および cycfg_gatt_db.h ファイルには、Bluetooth® Configurator ツールを使用して前のステップで生成された Find Me Target プロファイルの Bluetooth® Low Energy GATT データベース定義が含まれています。GATT データベースは、Bluetooth®スタックとアプリケーションコードの両方からアクセスされます。スタックは、属性ハンドル, UUID, および属性アクセス許可に直接アクセスして、Bluetooth®イベントの一部を処理します。アプリケーションコードは GATT データベースにアクセスして、属性の読み出し/書き込み操作を実行します。関連するデータベース構造を以下に示します。

- gatt_database[]: この配列には、属性ハンドル, タイプ, および権限が含まれています。この配列には、以下で説明するように個別の配列として保持される実際の属性値は持たないことに注意してください。
- GATT Value Arrays: 属性値を含む実際の GATT データベースは、*cycfg_gatt_db.c* の GATT Initial Value Arrays セクションで一連の uint8_t 配列として宣言されます。これらの配列は、*cycfg_gatt_db.h* ファイルでアプリケーションコードにアクセスするための外部変数としても公開されます。FMP ターゲットアプリケーションには、app_gap_device_name[], app_gap_appearance[], および app_ias_alert_level[]という名前で定義されたこれらの配列があります。app_ias_alert_level[]は、クライアントがアラートレベルを設定するために書き込む IAS サービスに対応するアラートレベル特性です。この属性への実際の書き込みは、アプリケーションコードによって実行されます。
- app_gatt_db_ext_attr_tbl[]: この構造体の配列は、gatt_database[]で定義されている属性ハンドルの GATT value arrays へのマッピングを伝達する GATT ルックアップテーブルです。アプリケーションコードはこのルックアップテーブルを使用して、実際の GATT arrays に対して属性の読み出し/書き込み操作を実行します。

5.8.2　Bluetooth®スタック設定パラメーター

app_bt_cfg.c および app_bt_cfg.h ファイルには、デバイス名 (app_gap_device_name), コアスタック設定パラメーター (wiced_bt_cfg_settings[]), およびスタックで使用されるメモリーバッファプール (wiced_bt_cfg_buf_pools[]) などのランタイム Bluetooth®スタック設定パラメーターが含まれています。このアプリケーションノートの範囲では、これらのパラメーターについては説明しません。ただし、これらのパラメーターについては、ソースファイルのコメントを参照してください。

app_gap_device_name 変数で定義されたデバイス名は、接続を確立するデバイスを識別するためにピアデバイス側で使用されるものであることに注意してください (この場合は「Find Me Target」)。デバイス名は、GATT 設定の Bluetooth® コンフィギュレーターで設定できます。

5.8.3 ユーザーアプリケーションコードの入力

main.c ファイルには、application_start()関数が含まれます。この関数は、デバイスの初期化が完了した後にユーザーアプリケーションコードを実行するためのエントリポイントです。このサンプルコードでは、この関数は 2 つのことを行います。

- PUART シリアルポートをデバッグ UART として選択して、トレースメッセージを表示し、WICED_BT_TRACE 関数を使用してデバッグ UART に起動メッセージを出力してください。WICED_BT_TRACE 関数は、スプリントタイプのフォーマットを使用してメッセージを送信するために使用されます。
- wiced_bt_stack_init()を呼び出して、Bluetooth®スタックマネジメントコールバック関数を登録してください。スタックマネジメントコールバック関数は、通常、Bluetooth®イベントに基づいてアプリケーションの残りの部分を制御します。通常、application_start()関数では、最小限のアプリケーション初期化のみが行われます。Bluetooth®スタックが有効になると、ほとんどのアプリケーション初期化はスタックコールバック関数で行われます。スタックコールバック関数 app_bt_management_callback は、app_bt_event_handler.c で定義されています。コールバック関数は、特定のイベントが発生したときに別の関数によって呼び出される関数です。

5.8.4 Bluetooth®スタックイベント

app_bt_event_handler.c および app_bt_event_handler.h ファイルには、スタックによって生成されるさまざまなタイプのイベントを処理するためのアプリケーションコードロジックが含まれます。大まかに、次の 2 つのカテゴリのイベントを処理する必要があります。

- Bluetooth®スタックマネジメントイベント
- GATT イベント

5.8.4.1 Bluetooth®スタックマネジメントイベント

コールバック関数 app_bt_management_callback は、スタック有効化, アドバタイズメントの状態変更, ペアリングなどのセキュリティ関連イベント, およびキー交換などのイベントを処理します。このコールバック関数は、*main.c* の application_start()関数の一部として登録されます。マネジメントイベントのリストについては、*wiced_bt_dev.h* の wiced_bt_management_evt_t 定義を参照してください。アプリケーションコードがすべてのマネジメントイベントを処理する必要はありません。処理されるイベントは、アプリケーションの要件によって異なります。**Figure 30** に、このサンプルコードのスタックマネジメントイベントハンドラーの実行ロジックを示します。

2 つのマネジメントイベント (BTM_ENABLED_EVT および BTM_BLE_ADVERT_STATE_CHANGED_EVT) のみがスタックマネジメントコールバック関数で扱われることを、この図は示します。

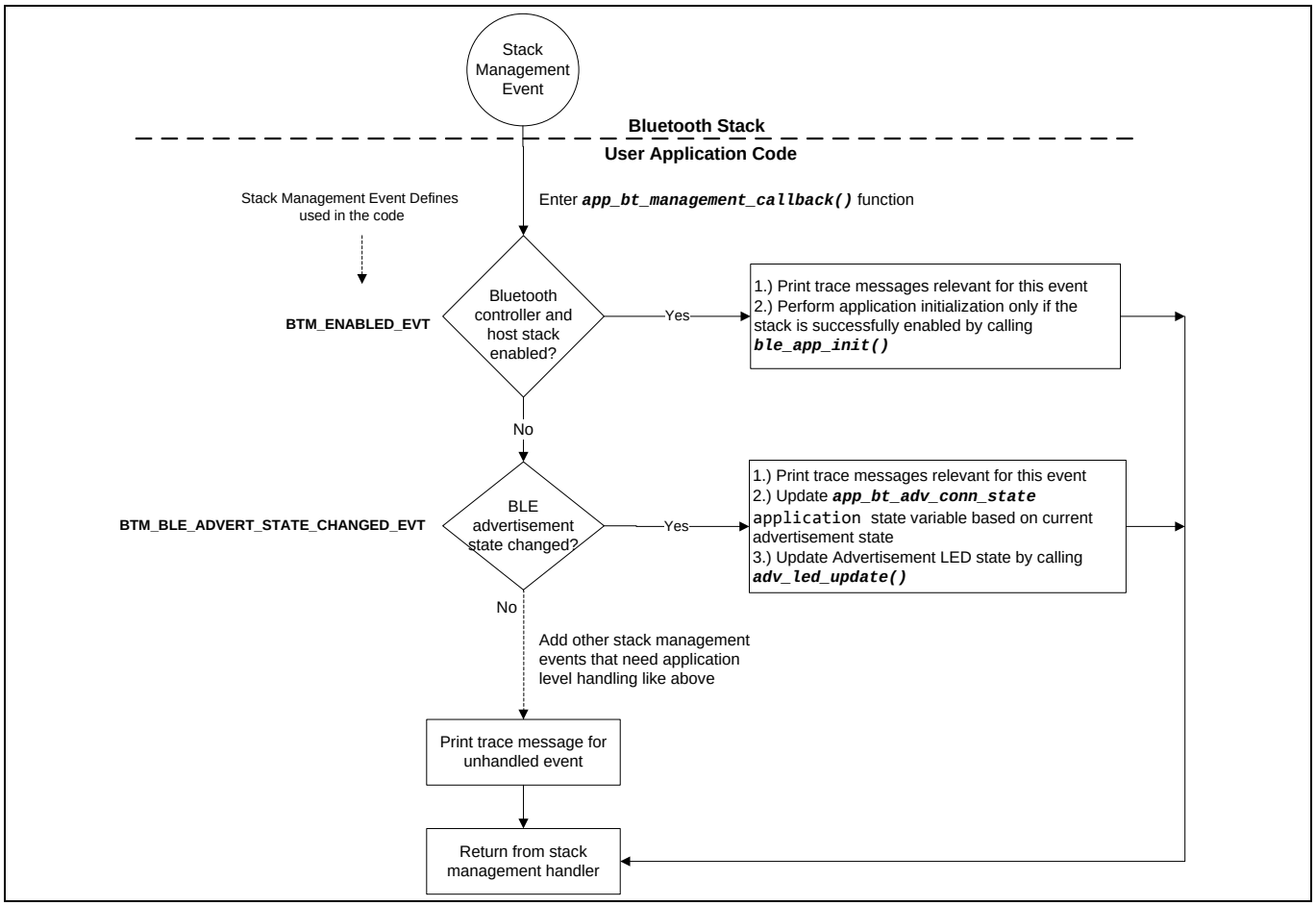

Figure 30　Bluetooth®スタックマネジメントイベントハンドラーの機能フロー

この時点で、BTM_ENABLED_EVT について説明することが適切です。これは、すべての CYW20820 ベースのアプリケーションで処理する必要がある重要なマネジメントイベントであり、Bluetooth®スタックが有効になっていることを示します。すべてのアプリケーションコードの初期化は、**Figure 30** に示すように、ble_app_init()関数を呼び出して Bluetooth®スタックが正常に有効化された後にのみ行われます。

app_bt_event_handler.c で定義される ble_app_init()関数は、以下に示す初期化タスクを実行します。作成する CYW20820 ベースのアプリケーションでは、この関数に必要な初期化コードを追加する必要があります。

- app_user_interface.c で定義されている app_user_interface_init()を呼び出して、ユーザーインターフェース (LED) ロジックを初期化してください。
- wiced_bt_set_pairable_mode()を呼び出してペアリングを無効にしてください。このアプリケーションでは、ペアリング機能は使用されません。
- *app_bt_event_handler.c* で定義されている ble_app_set_advertisement_data()を呼び出して、アドバタイズパケットデータを設定してください。サンプルコードのこの関数定義を見て、アドバタイズパケットの要素を設定する方法を理解してください。
- wiced_bt_gatt_register()を呼び出して、GATT イベントを処理するコールバック関数 (ble_app_gatt_event_handler()) を登録してください。
- wiced_bt_gatt_db_init()を呼び出して、cycfg_gatt_db.c で定義されている GATT データベース (gatt_database) を初期化してください。
- 初期化プロセスの最後のステップとして、デバイスは関数 wiced_bt_start_advertisements()を呼び出してアドバタイズを開始します。

5.8.4.2 GATT イベント

ble_app_gatt_event_handler()関数は、接続や属性要求イベントなどの GATT イベントを処理します。この関数は、app_bt_event_handler.c の ble_app_init()関数から wiced_bt_gatt_register()を呼び出して登録されます。GATT イベントのリストは、wiced_bt_gatt.h の wiced_bt_gatt_evt_t 定義を参照してください。アプリケーションコードがすべての GATT イベントを処理する必要はありません。処理されるイベントは、アプリケーションの要件によって異なります。**Figure 31** に、このサンプルコードの GATT イベントハンドラーの実行ロジックを示します。この図は、関数で 2 つの GATT イベント (GATT_CONNECTION_STATUS_EVT と GATT_ATTRIBUTE_REQUEST_EVT) のみが処理されることを示します。

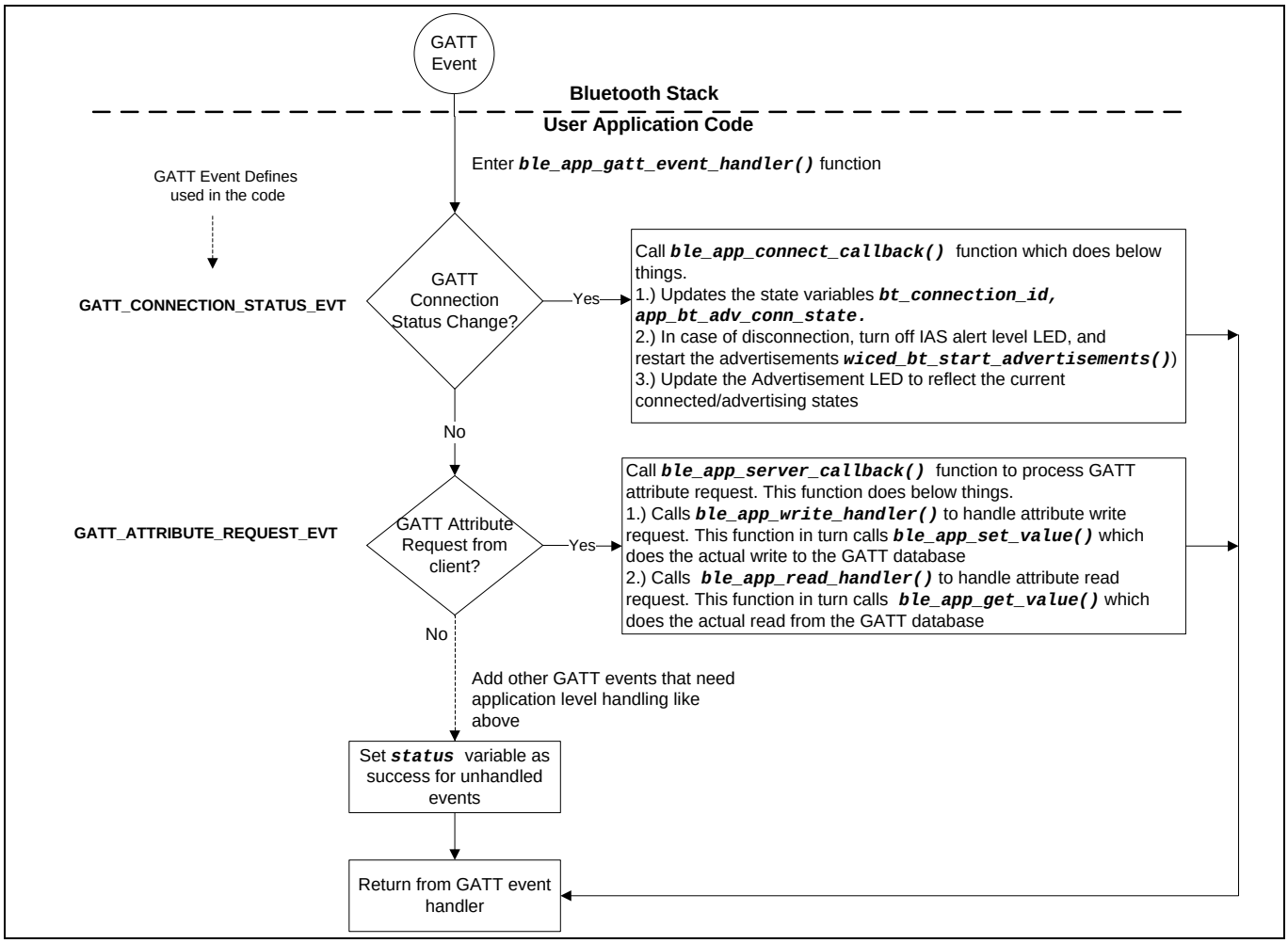

Figure 31 GATT イベントハンドラー

この時点で、GATT_ATTRIBUTE_REQUEST_EVT について議論することが適切です。このイベントは、GATT 属性の読み出し/書き込み操作を処理するために使用されます。**Figure 32** に、読み出しまたは書き込み操作の場合に呼び出される関数に関する情報を示します。このサンプルコードでは、Find Me Locator が CYW20820 デバイスの IAS アラートレベル特性を更新すると、GATT_ATTRIBUTE_REQUEST_EVT がトリガーされ、属性の書き込み要求に関連する一連の関数が呼び出されます。書き込み操作の最後に、cycfg_gatt_db.c の GATT データベースの app_ias_alert_level[]関数が、Find Me Locator によって設定されたアラートレベルで更新され、LED がアラートレベルに適切に設定されます。

Figure 32 に、このアプリケーションのさまざまなスタックイベントの関数呼び出しのシーケンスをまとめた関数呼び出しチャートを示します。これらすべての関数 (adv_led_update()を除く) は、app_bt_event_handler.c で定義されます。これらの関数の実装の詳細を理解するには、ソースコードを参照してください。

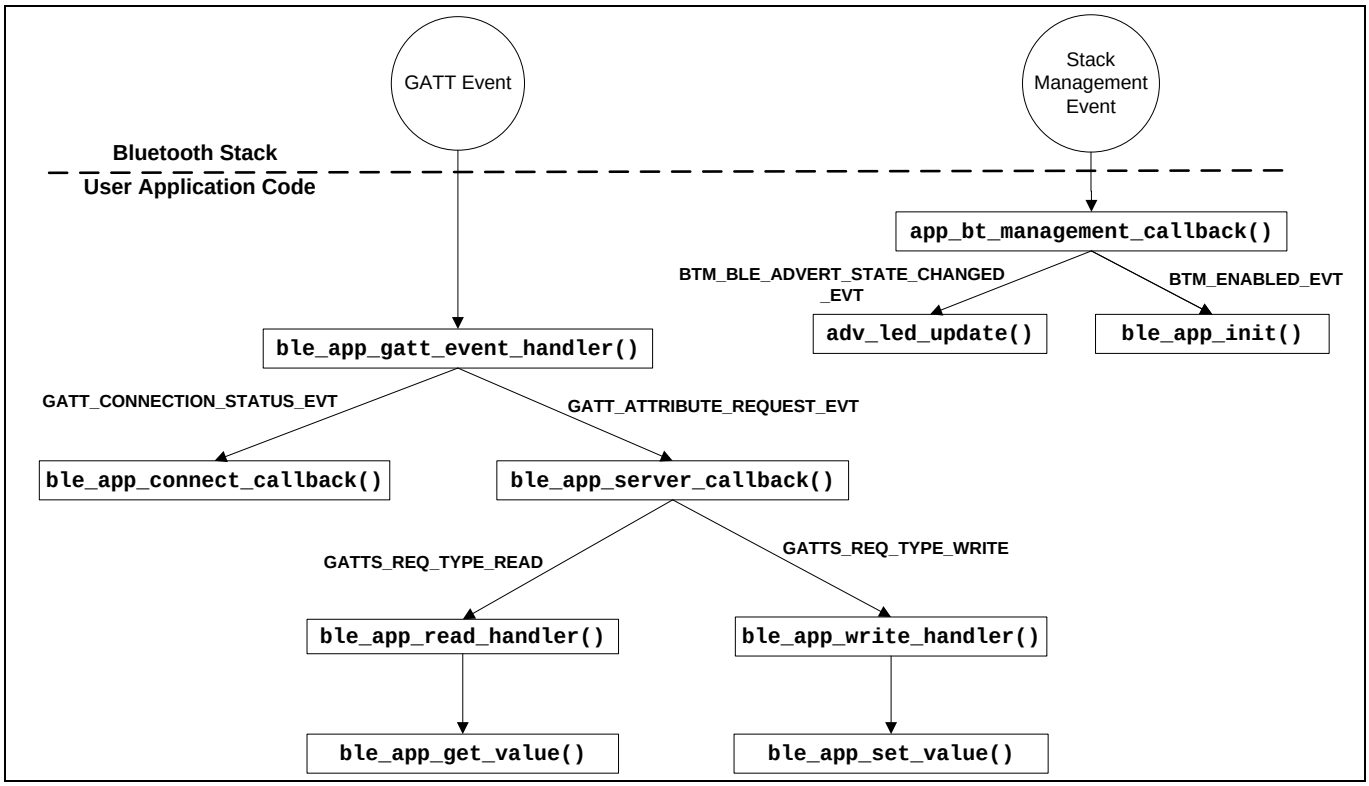

Figure 32　Bluetooth®スタックイベント関数呼び出しチャート

5.8.5　ユーザーインターフェースロジック

app_user_interface.c および app_user_interface.h ファイルには、ユーザーインターフェースロジックを処理するためのアプリケーションコードが含まれます。このデザインでは、以下のようにユーザーインターフェースに 2 つの LED を使用します。

- キットの LED1 (黄色の LED) は、Bluetooth® Low Energy 周辺機器のアドバタイズ/接続状態を示します。LED1 は、デバイスがアドバタイズしていないときはオフ状態、アドバタイズ中は点滅状態、ピアデバイスに接続されているときは常にオン状態です。実装の詳細については、adv_led_update() 関数を参照してください。グローバル状態変数 app_bt_adv_conn_state は、LED 状態を更新するために使用されます。adv_led_update()関数は、アプリケーションコードの 2 つの場所から呼び出されます。
  - app_bt_management_callback()関数は、通知状態が変化すると LED1 を更新します (スタックマネジメントイベント BTM_BLE_ADVERT_STATE_CHANGED_EVT)
  - ble_app_connect_callback()関数は、接続状態が変化すると LED1 を更新します (GATT イベント GATT_CONNECTION_STATUS_EVT)
- キットの LED2 (赤色 LED) は、デバイスがピアデバイスに接続されているときの IAS アラートの特性レベルを示します。ピアデバイスに接続されている場合、LED2 は、低アラートの場合はオフ状態、中アラートの場合は点滅状態、高アラートの場合はオン状態になります。デバイスがピアデバイスに接続されていない場合、LED2 はオフの状態です。実装の詳細については、ias_led_update()関数を参照してください。adv_led_update()関数は、アプリケーションコードの 2 つの場所から呼び出されます。
  - クライアント側から IAS アラートレベル特性への属性書き込み要求が行われると、ble_app_set_value()関数は LED2 を更新します。
  - ble_app_connect_callback()関数は、切断が発生したときに LED2 をオフ状態に駆動します (GATT イベント GATT_CONNECTION_STATUS_EVT)

5.9 パート 4: 設計のビルド, プログラミング, およびテスト

ここでは、CYW920820M2EVB-01 キットでアプリケーションをビルドし、CYW20820 デバイスをプログラムする方法を示します。また、LightBlue モバイルアプリを使用して Find Me Profile Bluetooth® Low Energy 設計をテストし、PUART シリアルインターフェースを使用して Bluetooth®スタックとアプリケーショントレースメッセージを表示する方法についても説明します。

この時点では、このアプリケーションノートの前の手順に従って Find Me Profile アプリケーションを開発していることを前提とします。

パス	「Using CE directly」 パス (既存のサンプルコード (CE) を直接評価)	「Working from Scratch」 パス (既存のサンプルコード (CE) を参照としてのみ使用)
アクション	このセクションのすべての手順を実行してください	このセクションのすべての手順を実行してください

付属の USB ケーブルを使用してキットを PC に接続してください。

1. キットの USB シリアルインターフェースは、CYW20820 デバイスの 2 つの UART インターフェース (WICED HCI UART と WICED ペリフェラル UART (PUART)) へのアクセスを提供します。HCI UART インターフェースは、このサンプルコードのアプリケーションコードをダウンロードするためにのみ使用されます。PUART インターフェースは、Bluetooth®スタックとアプリケーショントレースメッセージを出力します。お気に入りのシリアルターミナルアプリケーションを使用して、PUART シリアルポートに接続してください。次の設定を使用して、シリアルポートにアクセスするようにターミナルアプリケーションを設定してください。
2. ボーレート: 115200 bps; データ: 8 ビット; パリティ: なし; 停止: 1 ビット; フロー制御 – なし; 受信データの改行: 改行 (LF) または自動設定
3. アプリケーションのビルドとプログラミング: プロジェクトエクスプローラーで、**<App Name>_LE_Find_Me** プロジェクトを選択してください。クイックパネルで、**Launches** セクションまでスクロールして、**Figure 33** に示すように **<App Name> Attach_KitProg3_MiniProg4** 設定をクリックしてください。

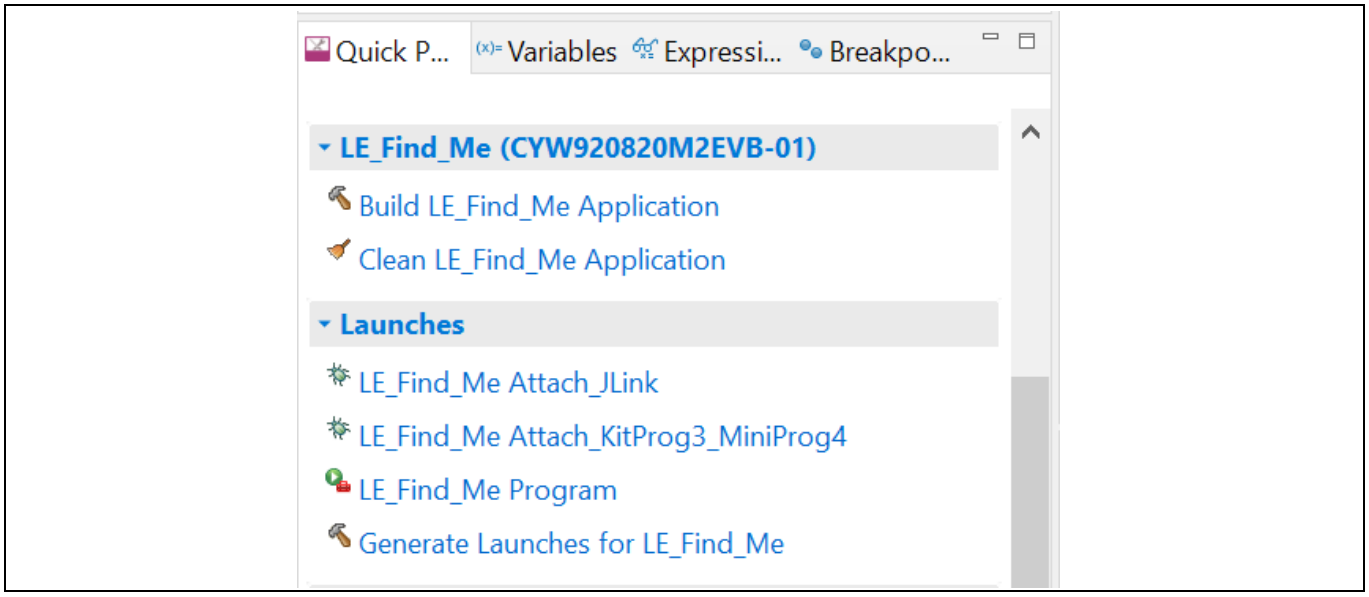

Figure 33 ModusToolbox™からの CYW20820 デバイスのプログラミング

Note: ダウンロードが失敗した場合、以前にロードされたアプリケーションがプログラミングを妨げている可能性があります。例えば、アプリケーションがダウンロードプロセスで検出されないカスタムボーレートを使用している場合や、デバイスが低電力モードにな

*っている場合があります。その場合、ボードをリカバリーモードにして、IDE からプログラミング操作を再試行する必要がある場合があります。リカバリーモードに入るには、最初に **Recover** ボタン(SW1) を押して保持し、それから **Reset** ボタン(SW2) を押し、**Reset** ボタン(SW2) を離し、次に **Recover** ボタン(SW1) を離してください。*

Note: アプリケーションのビルドプロセスでエラーが発生した場合、IDE コンソールウィンドウのエラーメッセージを読み、このドキュメントの関連する前の手順に戻って、これらの手順のいずれかを実行していないか、誤って実行していないかどうかを確認してください。

4. LightBlue モバイルアプリを使用してテストするには、以下の手順に従ってください (iOS の場合は **Figure 34**、Android の場合は **Figure 35** の同等の LightBlue アプリのスクリーンショットを参照してください)。

- Android または iOS デバイスで Bluetooth®をオンにしてください。
- LightBlue アプリケーションを起動してください。
- CYW920820M2EVB-01 キットのリセットスイッチを押して、アドバタイズメントの送信を開始してください。黄色の LED (LED1) が点滅し始め、広告が開始されたことを示します。接続が確立されていない場合、アドバタイズは 90 秒後に停止します。
- LightBlue アプリのホーム画面を下にスワイプして、Bluetooth® Low Energy ペリフェラルのスキャンを開始してください。デバイスが LightBlue アプリのホーム画面に表示されます。デバイスを選択して、Bluetooth® Low Energy 接続を確立します。接続が確立されると、黄色の LED (LED1) が点滅状態から常時オン状態に変わります。
- 「Alert Level」オプションを選択してください。
- [Find Me Profile]画面でアラートレベルの値を選択してください。デバイスの赤色 LED (LED2) の状態がアラートレベルに基づいて変化することを確認してください。

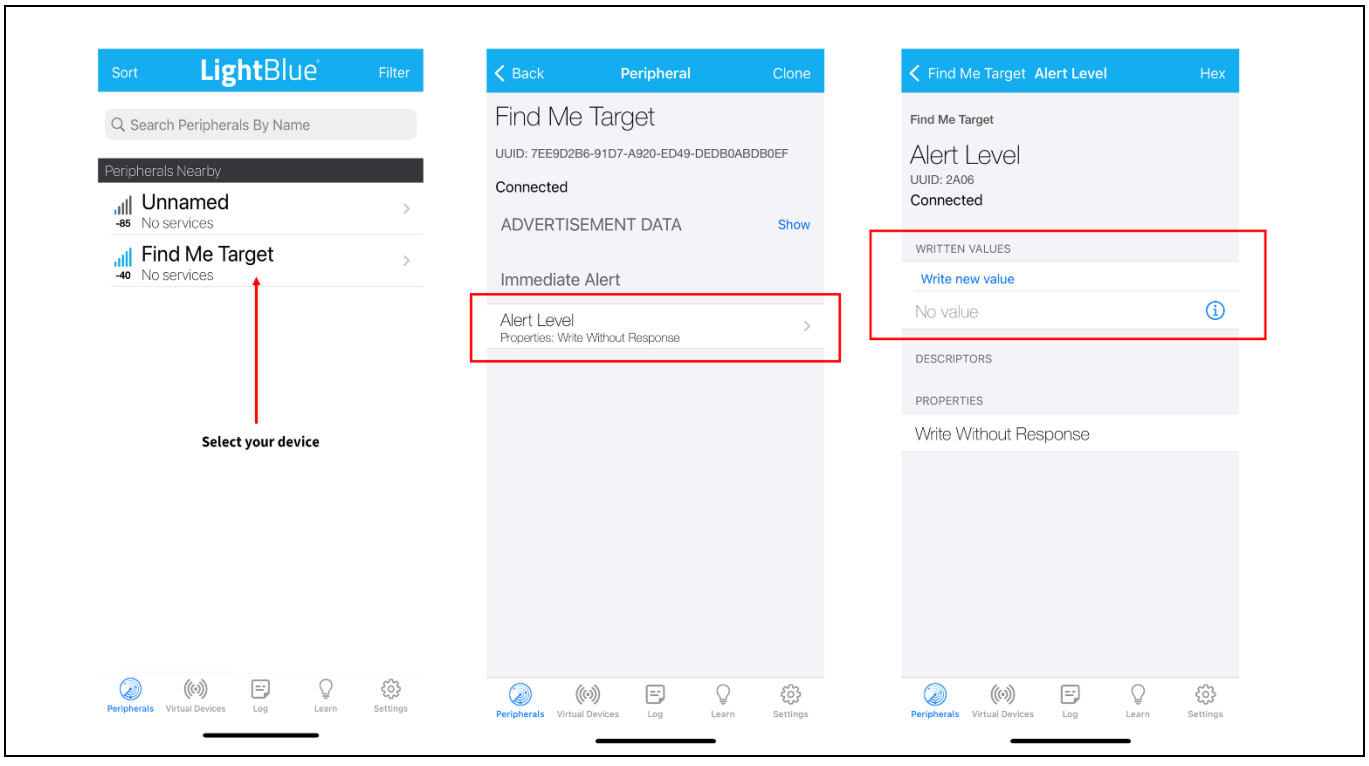

Figure 34 iOS 上の LightBlue アプリを使用したテスト

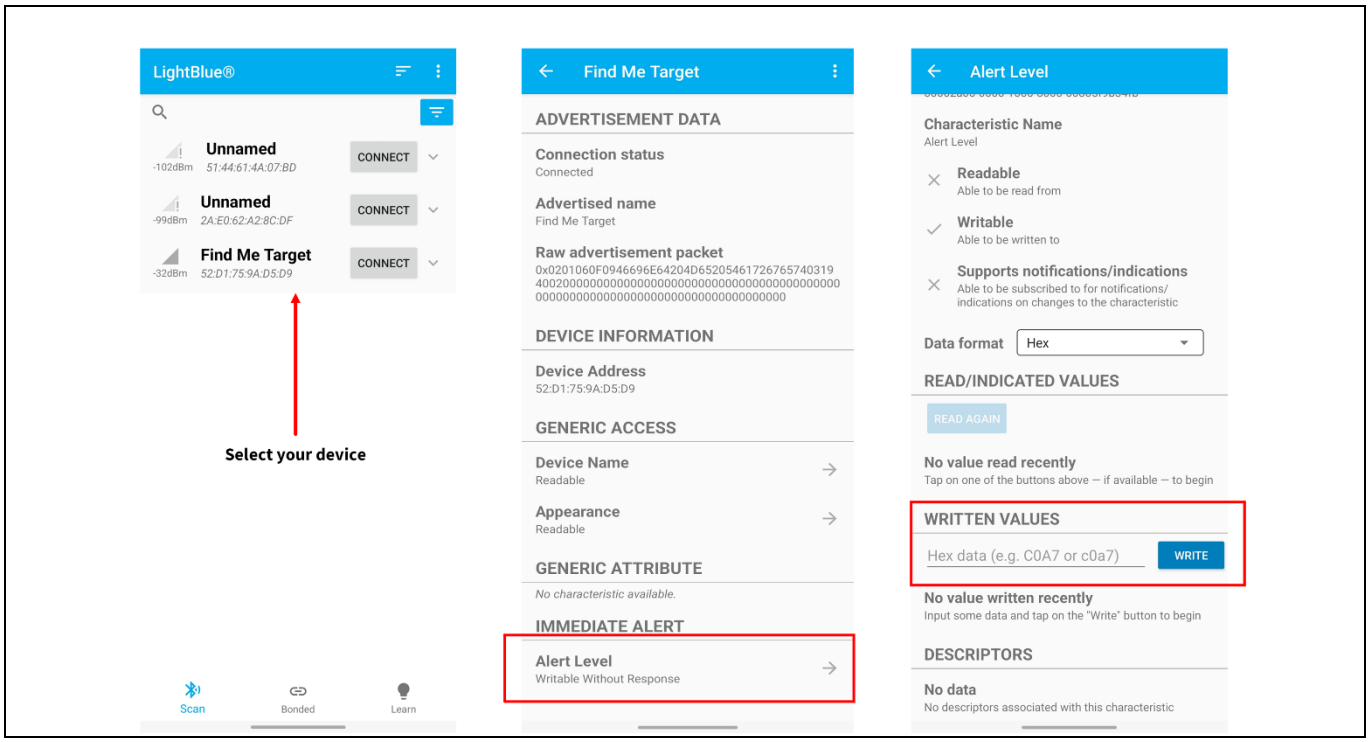

Figure 35　Android 上の LightBlue アプリを使用したテスト

5. **Figure 36** に示すように、PUART シリアルポートを使用して、ターミナルウィンドウに Bluetooth®スタックとアプリケーショントレースメッセージを表示してください。

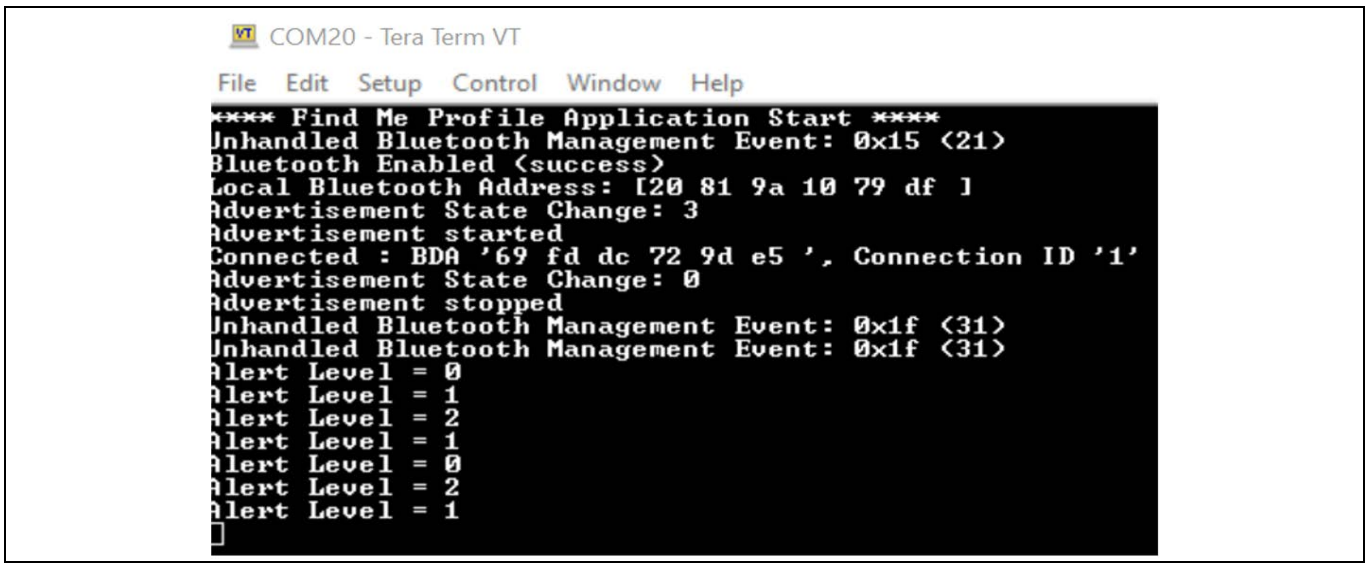

Figure 36　WICED PUART シリアルポートのログメッセージ

これで、Eclipse IDE for ModusToolbox™を使用した CYW208xx デバイス用のシンプルな Bluetooth® Low Energy アプリケーションの開発に成功しました。技術文書、追加のサンプルコードを含む CYW20820 デバイスの詳細については、**参考資料**セクションを参照してください。

6 まとめ

このアプリケーションノートでは、CYW208xx Bluetooth® MCU デバイスアーキテクチャ、関連する開発ツール、および ModusToolbox™を使用して CYW208xx のシンプルな Bluetooth® Low Energy アプリケーションを作成する手順について説明しました。CYW208xx は、Bluetooth® 5.0 に準拠したスタンドアロンのベースバンドプロセッサで、Bluetooth® Low Energy とクラシック Bluetooth®の両方をサポートする 2.4 GHz トランシーバーを統合しています。このデバイスは、オーディオ (ソース), センサー (医療, 家庭, セキュリティ), HID, リモートコントロール機能, およびその他の IoT アプリケーションのホストでの使用を目的としています。

インフィニオンは、エンドアプリケーションの要件を満たす CYW20820 ベースの Bluetooth®アプリケーションを迅速に開発するのに役立つ豊富なサンプルコード, アプリケーションノート, およびその他の技術ドキュメントを提供しています。CYW20820 デバイスについてさらに学習し、Bluetooth®アプリケーションを開発するには、**参考資料**のセクションを参照してください。

参考資料

[1] アプリケーションノート

- **ModusToolbox™ 2.4 User Guide**

[2] サンプルコード

- **Code Examples for ModusToolbox™ Software** – ModusToolbox™ IDE を使用したサンプルコードの包括的なコレクションについては、このサンプルコードにアクセスしてください。

[3] デバイス データシート

- **CYW20835 device datasheet**
- **CYW20819 device datasheet**
- **CYW20820 device datasheet**

[4] 開発キット

- **CYW920820M2EVB-01 evaluation kit**
- **CYW920819M2EVB-01 evaluation kit**
- **CYW920835M2EVB-01 evaluation kit**

[5] ツールのドキュメント

- **Eclipse IDE for ModusToolbox™ -** IoT デザイナー向けのインフィニオン IDE

改訂履歴

Document version	Date of release	Description of changes
**	2020-10-20	本版は英語版 002-25684 Rev. **について、CYPRESS DEVELOPER COMMUNITY の参画者によって日本語に翻訳されたドキュメントです。
*A	2021-07-21	テンプレートの変更を実施。 これは英語版 002-25684 Rev. *A を翻訳した日本語版 Rev. *A です。
*B	2022-12-20	これは英語版 002-25684 Rev. *B を翻訳した日本語版 Rev. *B です。

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

Edition 2022-12-20
Published by
Infineon Technologies AG
81726 Munich, Germany

© 2022 Infineon Technologies AG.
All Rights Reserved.

Do you have a question about this document?
Go to www.infineon.com/support

Document reference
002-31588 Rev. *B

重要事項

本手引書に記載された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。本製品の使用の前に、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。