PSoC™ 4 CY8C4xxx-BL MCU with AIROC™ Bluetooth® LE 入门

关于本文档

范围及目的

AN91267 为您介绍了 PSoC™ 4 with AIROC™ Bluetooth® LE (PSoC™ 4 CY8C4xxx-BL MCU with AIROC™ Bluetooth® LE)，它是一个基于 Arm® Cortex®-M0 的可编程片上系统 (PSoC™)，该 PSoC™集成了 Bluetooth® LE 无线射频系统。您可以通过本应用笔记了解 PSoC™ 4 with AIROC™ Bluetooth® LE 架构和开发工具，并能够学会如何使用 PSoC™ Creator (即为 PSoC™ 4 with AIROC™ Bluetooth® LE 的开发工具) 轻松创建 Bluetooth® LE 设计。此外，本应用笔记还为您提供了多项资源，从而可以加快您的设计开发，并能加深对有关 PSoC™ 4 with AIROC™ Bluetooth® LE 知识的学习。

Intended audience

本文档适用于使用配备 AIROC™ Bluetooth® LE 的 PSoC™ 4 新用户。

相关器件系列

CY8C41x7-BL、CY8C42x7-BL、CY8C41x8-BL、CY8C42x8-BL

软件版本

PSoC™ Creator 4.4

更多代码示例？我们倾听到了你的声音。

要查看持续增长的成百上千的 PSoC™代码示例，请访问**代码示例网页**。您也可以在**这里**浏览 PSoC™视频库。

目录

关于本文档 1
目录 1
1 简介 4
1.1 先决条件 4
1.1.1 硬件 4
1.1.2 软件 4
2 PSoC™ 4 with AIROC™ Bluetooth® LE 设计资源 5
2.1 PSoC™ Creator 5
2.2 PSoC™ Creator 帮助 6
2.3 代码示例 7
3 PSoC™ 4 with AIROC™ Bluetooth® LE 特性 9
4 PSoC™的性能比 Bluetooth® LE MCU 的更优越 11
5 Bluetooth® LE 概述 13
5.1 Bluetooth® LE 连接的建立程序 14

目录

5.2 GATT 数据格式 15
5.3 Bluetooth® LE 配置文件 17
5.4 Bluetooth® LE 组件 19
5.4.1 特性 19
5.4.2 配置 19
6 PSoC™ 4 with AIROC™ Bluetooth® LE 开发设置 22
7 我的第一个 PSoC™ 4 with AIROC™ Bluetooth® LE 设计 24
7.1 设计概述 24
7.2 先决条件 25
7.3 第一步：构建设计 25
7.4 第二步：编写固件 36
7.4.1 系统初始化 36
7.4.2 Bluetooth® LE 堆栈事件处理程序 38
7.4.3 Bluetooth® LE 特定服务事件处理程序 41
7.4.4 主循环和低功耗操作 42
7.5 第三步：编程器件 44
7.6 第四步：测试您的设计 46
7.6.1 使用 CySmart 手机应用 46
7.6.2 使用 CySmart 电脑主机仿真工具 47
8 总结 51
9 附录 A：Bluetooth® LE 器件系列的比较 52
10 附录 B：Infineon 专业术语 55
11 附录 C：Infineon Bluetooth® LE 开发工具 56
11.1 CySmart 主机仿真工具 56
11.2 手机版的 CySmart 应用 58
12 附录 D：PSoC™ 4 with AIROC™ Bluetooth® LE 器件 60
12.1 Bluetooth® Low Energy 子系统 (BLESS) 60
12.1.1 深度睡眠模式 60
12.1.2 睡眠模式 60
12.1.3 空闲模式 60
12.1.4 发送模式 60
12.1.5 接收模式 60
12.2 Arm® Cortex®-M0、存储器和 DMA 61
12.3 可编程的数字外设 61
12.3.1 可编程 SCB 61
12.3.2 可编程 TCPWM 61
12.3.3 通用数字模块 61
12.3.4 应用 62
12.4 可编程模拟模块 62
12.4.1 微型的连续时间模块 (CTBm) 63
12.4.2 带有硬件定序器的 SAR ADC 63
12.4.3 低功耗比较器 63
12.4.4 电容式触摸感应 (CapSense™) 63
12.4.5 段式 LCD 直接驱动程序 63
12.4.6 应用 64
12.5 系统资源 64
12.5.1 低-漏功耗模式 64
12.5.2 电源供应及监测 65
12.5.3 时钟系统 65

目录

12.5.4 器件安全性 66
12.6 可编程 GPIO 66
13 附录 E：Bluetooth® LE 协议 67
13.1 概况 67
13.2 物理层 (PHY) 67
13.3 链路层 (LL) 67
13.4 主机控制接口 (HCI) 68
13.5 逻辑链路控制及适配协议 (L2CAP) 68
13.6 安全管理器 (SM) 69
13.7 属性协议 (ATT) 69
13.7.1 属性层次结构 70
13.7.2 属性操作 72
13.8 通用属性配置文件 (GATT) 73
13.9 通用访问配置文件 (GAP) 73
相关应用笔记 76
文档修订记录 77

1 简介

赛普拉斯的 PSoC™ 4 with AIROC™ Bluetooth® LE 器件是可编程的嵌入式片上系统，在单芯片中集成了 Bluetooth® Low Energy 无线射频系统、可编程模拟和数字外设功能、存储器以及 Arm® Cortex®-M0 微控制器。Bluetooth® LE 是由蓝牙技术联盟 (SIG) 制订的超低功耗无线标准，使用于短距离通信。它支持一个物理层、一个协议栈和各个应用使用情况。所有性能都是为了降低功耗而进行设计和优化的。

PSoC™ 4 with AIROC™ Bluetooth® LE 提供了一个经济实用且封装较小的解决方案，能够替代传统的 MCU 和 Bluetooth® LE 无线射频组合。可编程模拟和数字子系统为采用 PSoC™ Creator (基于原理图的设计工具，用于开发 PSoC™ 4 with AIROC™ Bluetooth® LE 应用) 的设计提供了优越的灵活性和调校能力。若想开发 Bluetooth® LE 应用，便不用了解 Bluetooth® LE 复杂协议栈的运作方式。Infineon 在 PSoC™ Creator 中提供了一个易于配置的基于图形用户界面 (GUI) 的免费 Bluetooth® LE 组件，这样有助于降低协议的复杂性。通过使用 GUI，可以轻松配置 Bluetooth® LE 组件，从而快速启动您的 Bluetooth® LE 设计。PSoC™ 4 with AIROC™ Bluetooth® LE 在电流消耗仅为 150 nA 时仍能保持 SRAM 内容和可编程逻辑，并能够从一个中断中唤醒。在深度睡眠模式下，它仅消耗 1.3 µA 的电流，但仍能够保持 Bluetooth® LE 连接有效。这些低功耗模式为电池供电的 Bluetooth® LE 设计 (如可穿戴的健身监测仪和无线传感器接口) 提供了一流的系统节能特性。

PSoC™ 4 with AIROC™ Bluetooth® LE 通过集成的 Balun 电路减少了天线匹配所需的外部组件数量，从而简化了 RF 电路板的设计。

Infineon PSoC™ 4 with AIROC™ Bluetooth® LE 中的电容式触摸感应特性，称为 CapSense™，能提供前所未有的信噪比和一流的防水性能，并支持各种类型的传感器，如按键、滑条、触控板和接近传感器。CapSense™ 用户界面在可穿戴电子设备 (如活动监测仪、健康和健身设备) 中的应用日益普遍。

本应用笔记介绍了有关 Bluetooth® LE 的基本信息，其中包括 PSoC™ 4 with AIROC™ Bluetooth® LE 器件的功能以及开发工具的概述。欲了解高级应用的开发信息，请参考**设计 Bluetooth® LE 应用**和**创建 Bluetooth® LE 自定义配置文件**应用笔记。

1.1 先决条件

开始进行前，请保证您已经具有开发套件并已经安装好了所需软件。

1.1.1 硬件

- **PSoC™ 4 Bluetooth® Low Energy 兼容 Pioneer 套件**
- 使用 Windows 7 或更高版本的 PC/笔记本电脑 (如果使用 **CySmart 主机仿真工具** PC 应用)
- 使用 Android 5 (或更高版本) 或 iOS 8 (或更高版本) 的手机 (如果使用 CySmart **iOS**/**Android** 应用)

1.1.2 软件

- **PSoC™ Creator 4.4**，内置 **PSoC™ Programmer 3.29**
- **CySmart 主机仿真工具**或 CySmart **iOS**/**Android** 应用

2 PSoC™ 4 with AIROC™ Bluetooth® LE 设计资源

Infineon 网站 **www.cypress.com** 上提供了大量资料，有助于您正确选择 PSoC™器件，并能够快速和有效地将器件集成到您的设计中。如果您是首次使用 PSoC™产品，那么建议您先阅读**附录 B：Infineon 专业术语**，了解常用的术语列表。有关使用资源的完整列表，请参考 **KBA86521 — 如何使用 PSoC™ 3、PSoC™ 4 和 PSoC™ 5LP 进行设计**。

下面提供了 PSoC™ 4 with AIROC™ Bluetooth® LE 的简要列表：

- **概述：PSoC™产品系列、PSoC™ RF 蓝图**
- **产品选型：PSoC™ 1、PSoC™ 3、PSoC™ 4 或 PSoC™ 5LP**。此外，**PSoC™ Creator** 还包含了一个器件选择工具。
- **数据手册：**介绍了 **PSoC™ 4XX7-BL** 和 **PSoC™ 4XX8-BL** 器件系列，并且提供了它们的相关电气规范。
- **应用笔记和代码示例**：包括从基本到高级的广泛主题。许多应用笔记还包含了代码示例。PSoC™ Creator 提供了其他代码示例 — 请参考**代码示例**一节。
- **技术参考手册 (TRM)：**详细说明了每个 PSoC™ 4 with AIROC™ Bluetooth® LE 器件系列中的架构和寄存器。
- **CapSense™设计指南：**了解如何使用 PSoC™ 4 with AIROC™ Bluetooth® LE 器件系列进行设计电容式触摸感应应用。
- **开发工具**
  - **CY8CKIT-042-BLE Bluetooth® Low Energy Pioneer 套件**是一款针对 Bluetooth® LE 器件系列的易用且价廉的开发平台。该套件包括用于与 Arduino™兼容的扩展板和 Digilent® Pmod™子卡的连接器。
  - 适用于 **Windows**、**iOS** 和 **Android** 的 CySmart Bluetooth® LE 主机仿真工具是一款易用的应用，通过它您可以测试和调试 Bluetooth® LE 外设应用。

  相关概述，请参考**附录 C：Infineon Bluetooth® LE 开发工具**。
- **培训视频：**一段视频比得上一万句介绍 — 下面将为您提供有助于开始进行第一个 Bluetooth® LE 设计的 PSoC™ 4 with AIROC™ Bluetooth® LE 视频：
  - **PSoC™ 4 with AIROC™ Bluetooth® LE 简介**
  - **开始了解 PSoC™ Creator**
  - **PSoC™ 4 Bluetooth® Low Energy Compliant Pioneer 套件**
  - **Bluetooth® Low Energy 简介**
  - **配置您的第一个 Bluetooth® Low Energy 设计**
  - **为您的第一个 Bluetooth® Low Energy 设计编程固件**
- 请访问我们的**技术支持**页面，了解与设计、兼容性、功能、时序、数据表和一般应用支持相关的问题。

2.1 PSoC™ Creator

PSoC™ Creator 是一个基于 Windows 的免费集成设计环境 (IDE)。通过它可以在 PSoC™ 4 with AIROC™ Bluetooth® LE 或 PRoC-BLE 器件中同时设计硬件和固件系统。如 **Figure 1** 所示，通过 PSoC™ Creator，您可以进行以下操作：

1. 将组件图标拖放到主设计工作区中，以进行您的硬件系统设计。
2. 同时对您的应用固件和 PSoC™硬件进行设计。
3. 使用配置工具来配置各组件。
4. 研究包含 100 多个组件的库。
5. 查看组件数据手册。

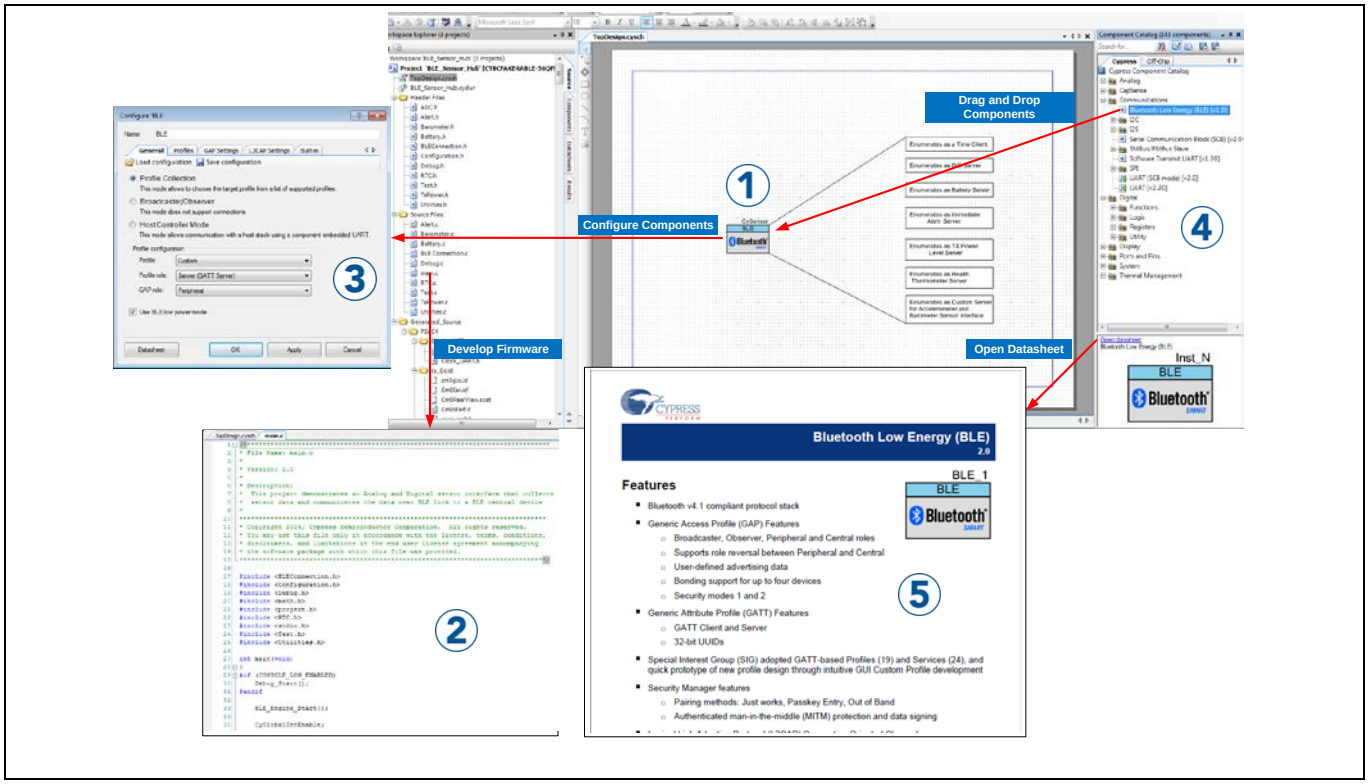


Figure 1 PSoC™ Creator 原理图和组件

2.2 PSoC™ Creator 帮助

请访问 **PSoC™ Creator** 主页，以下载并安装 PSoC™ Creator 的最新版本。然后，启动 PSoC™ Creator，并导航到下列各项：

- **快速入门指南**：依次选择 **Help** >**Documentation** > **Quick Start Guide**。本指南提供了与开发 PSoC™ Creator 项目有关的基本知识。
- **简单的组件代码示例**：依次选择 **File** > **Code Example**。这些代码示例展示了如何配置和使用 PSoC™ Creator 组件。
- **系统参考指南**：依次选择 **Help** > **System Reference Guides**。该指南列出并描述了 PSoC™ Creator 所提供的系统功能。
- 组件数据手册：右击组件，然后选择“Open Datasheet”项。请访问 **PSoC™ 4 with AIROC™ Bluetooth® LE 组件数据手册**网页，以获取 PSoC™ 4 with AIROC™ Bluetooth® LE 组件的所有数据手册列表。

2.3 代码示例

PSoC™ Creator 包含了多个代码示例。可以从 PSoC™ Creator 的 Start Page 上获取这些项目，如 **Figure 2** 所示。

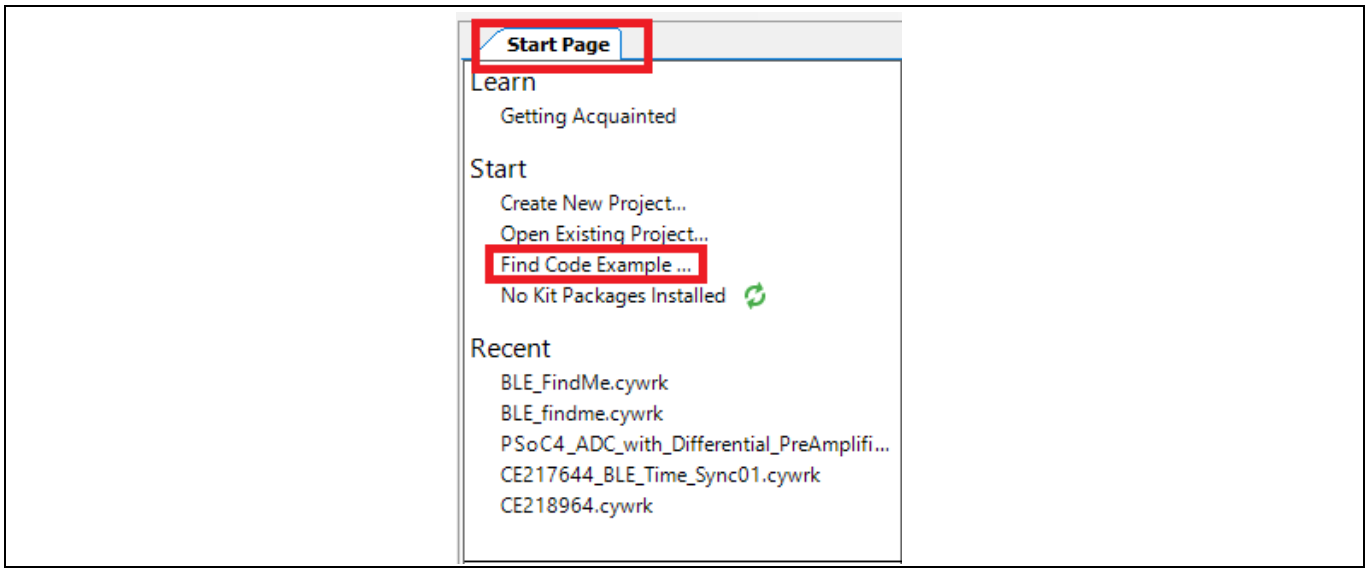


Figure 2 PSoC™ Creator 中的代码示例

通过向您提供完整的设计 (并非一个空白页)，这些代码示例可以加快您的设计进程。代码示例还介绍了如何将 PSoC™ Creator 组件使用于不同的应用中。

在 **Figure 3** 所示的 **Find Code Example** 对话框中，您可以选择以下选项：

- 根据架构或器件系列 (例如：PSoC™ 4、PSoC™ 4 with AIROC™ Bluetooth® LE、PRoC-BLE 等) 、类型或关键词等选项筛选示例。
- 从 Filter Options 的示例菜单中进行选择。共有 30 多个 Bluetooth® LE 代码示例供您参考，如 **Figure 3** 所示。
- 查看可选择的文档 (在 **Documentation** 选项卡中)
- 查看选中的代码示例。您可以复制该窗口中的代码，然后将其粘贴在您的项目内，从而加快代码开发过程；
- 或根据所选的项目创建一个新项目 (若需要可添加新的工作区) 。通过向您提供一个完整的基本设计，该方式可以加快您的设计进程。然后，您可以根据自己的应用要求来调整该设计。
- 除了 PSoC™ Creator 代码示例外，您还可以在该 GitHub 资料库中找到更多 Bluetooth® LE 参考示例。

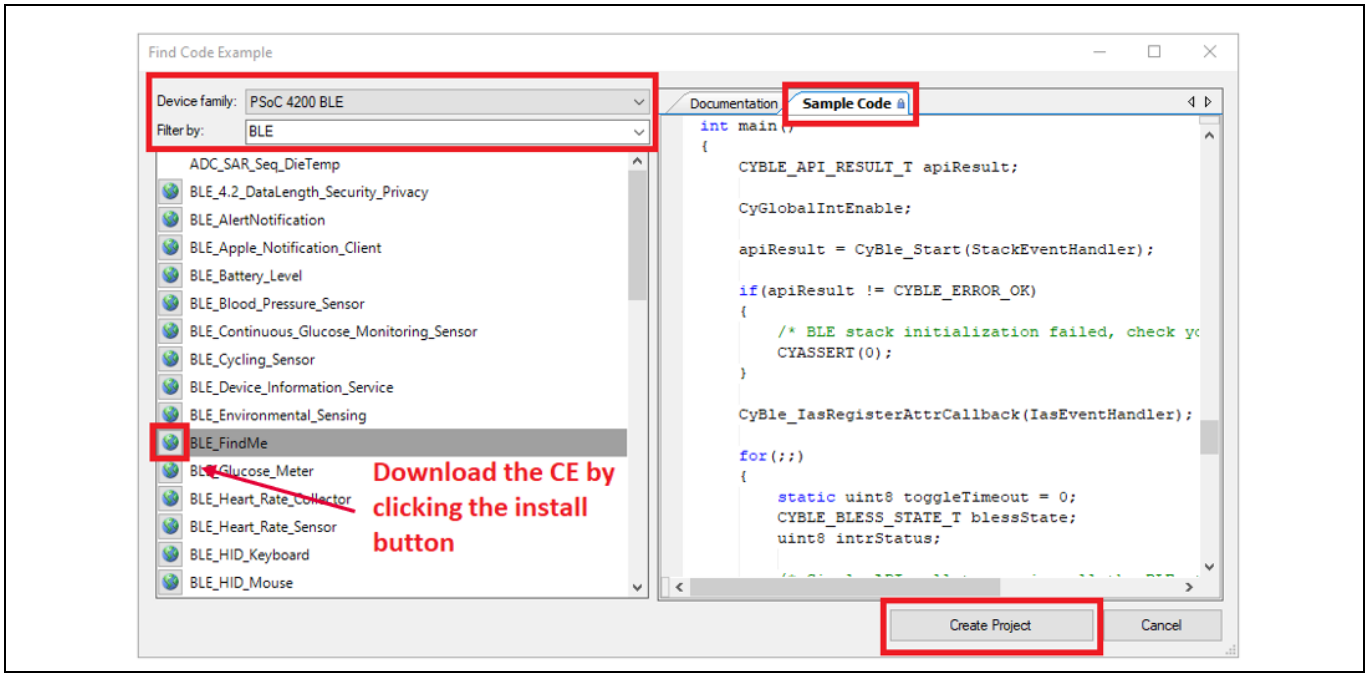


Figure 3　代码示例

3 PSoC™ 4 with AIROC™ Bluetooth® LE 特性

如 **Figure 4**，PSoC™ 4 with AIROC™ Bluetooth® LE 器件提供：

- **32 位 MCU 子系统**
  - 48 MHz Arm® Cortex®-M0 CPU
  - 最大支持 256 KB 的闪存和 32 KB 的 SRAM
- **具有 SmartSense™自动调校功能的 CapSense™**
  - 一个 Infineon 电容式 Sigma-Delta™ (CSD) 控制器，支持触控板功能
- **可编程模拟前端 (AFE)**
  - 四个运算放大器，可配置为 PGA、比较器、滤波器等
  - 一个 12 位 SAR ADC，支持 1 Msps 的采样率
- **可编程数字逻辑**
  - 四个通用数字模块 (UDB)： 自定义数字外设
  - 四个 TCPWM 模块，可配置为：16 位定时器、计数器或 PWM
  - 两个串行通信模块 (SCB)，可配置为：I2C 主/从设备、SPI 主/从设备或 UART
- **满足蓝牙 4.2 规范的 Bluetooth Smart 连接**
  - 集成了 Balun 的 2.4 GHz Bluetooth® LE 无线射频
  - 实现满足蓝牙 4.2 规范的控制器和主机

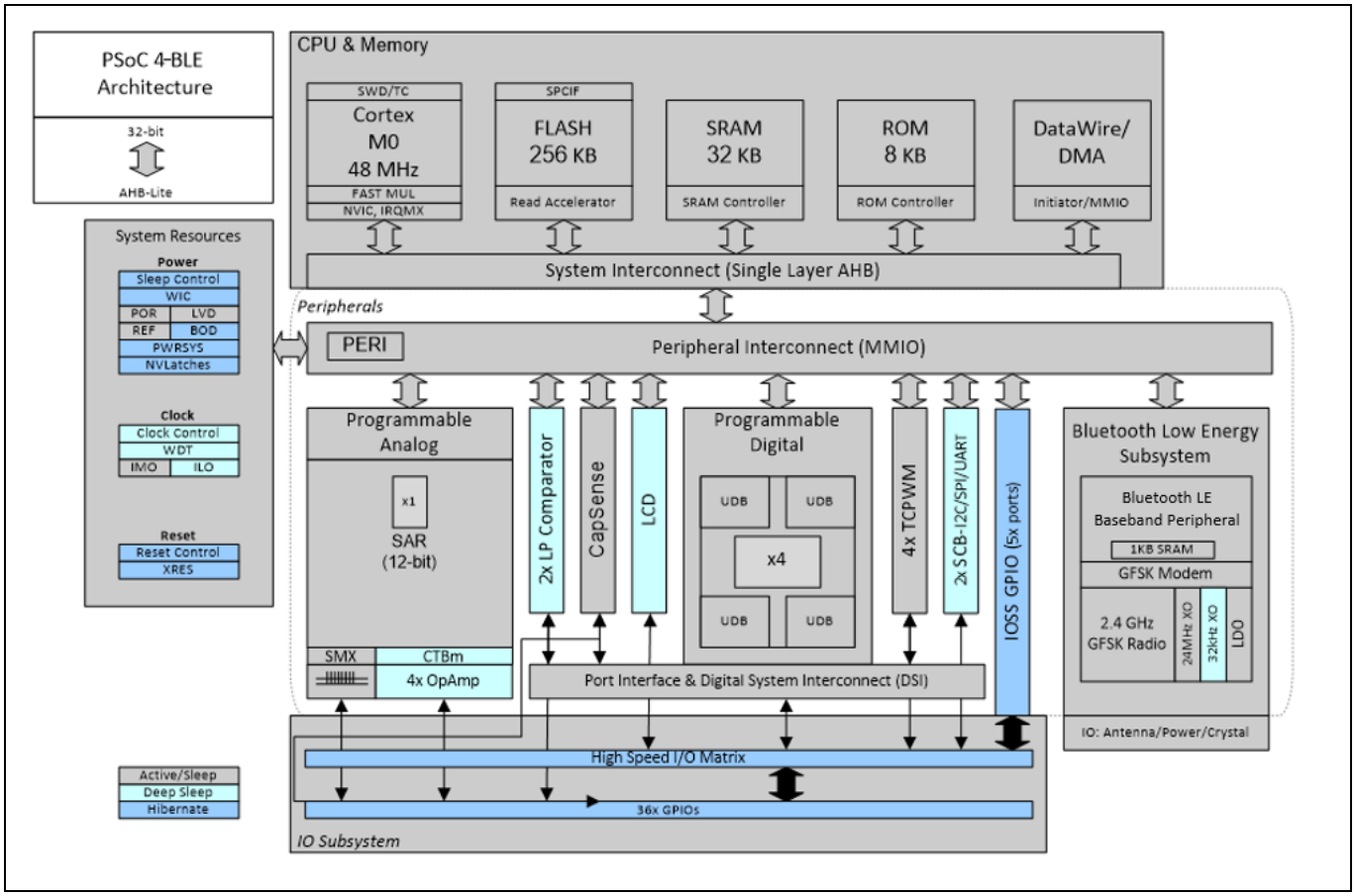


Figure 4 PSoC™ 4 with AIROC™ Bluetooth® LE 架构 (CY8C4248-BL)

PSoC™ 4 with AIROC™ Bluetooth® LE 特性

要想查看这些特性的简要说明，请参考**附录 D：PSoC™ 4 with AIROC™ Bluetooth® LE 器件**。更多有关信息，请参考 PSoC™ 4 with AIROC™ Bluetooth® LE 系列**数据手册**、**技术参考手册 (TRM)** 和应用笔记。**Figure 4** 显示了 CY8C4248-BL 器件系列所具有的各项特性。其他器件系列具备其中部分特性，请参考 **Table 2**。

4 PSoC™的性能比 Bluetooth® LE MCU 的更优越

Figure 5 显示了一款典型的 MCU，它包含一个 CPU (如：8051 或 Arm® Cortex®)，并且具有一系列外设功能 (如：ADC、DAC、UART、SPI、Bluetooth® LE) 和通用 I/O，这些部分均与 CPU 的寄存器接口相连。在 MCU 中，可以将 CPU 称为器件的“心脏” – 它监控着器件中从设置、数据传输到时序等所有活动。如果没有 CPU，那么该 MCU 便不能执行其性能。

Figure 6 显示的 PSoC™又存在差别。在 PSoC™中，CPU、模拟、数字和 I/O 都是同样重要的资源。*PSoC™的“心脏”是系统的互联和可编程性，并不是 CPU。*外设模拟和数字通过高度可编程信号矩阵和数据总线网络互相连接，用以创建符合您应用要求的自定义设计。*因此，您可以通过编程 PSoC™来模拟 MCU，但不能通过编程 MCU 来模拟 PSoC™。*

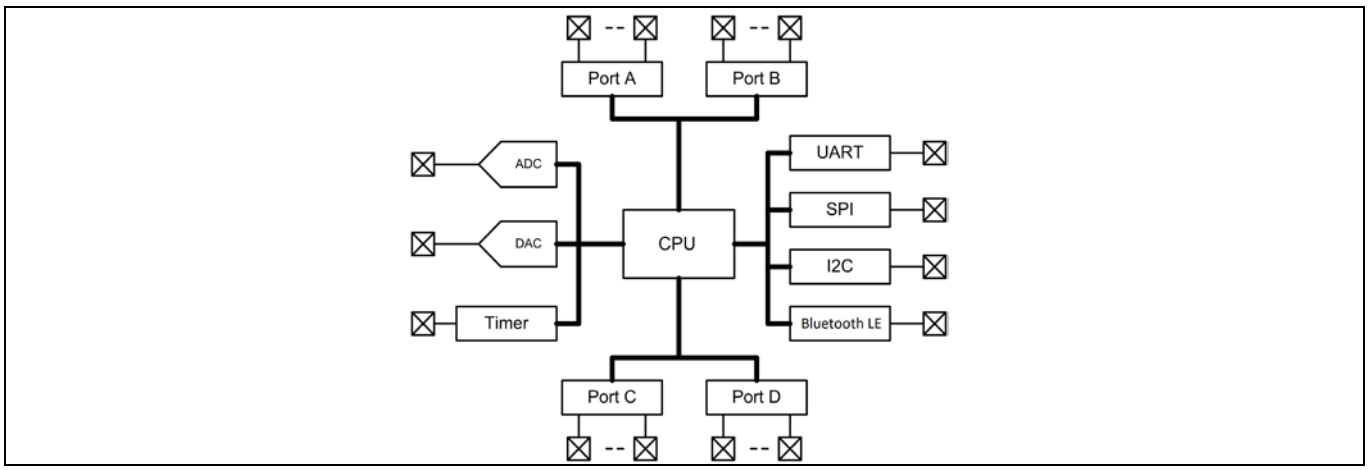


Figure 5 典型的 Bluetooth® LE MCU 框图

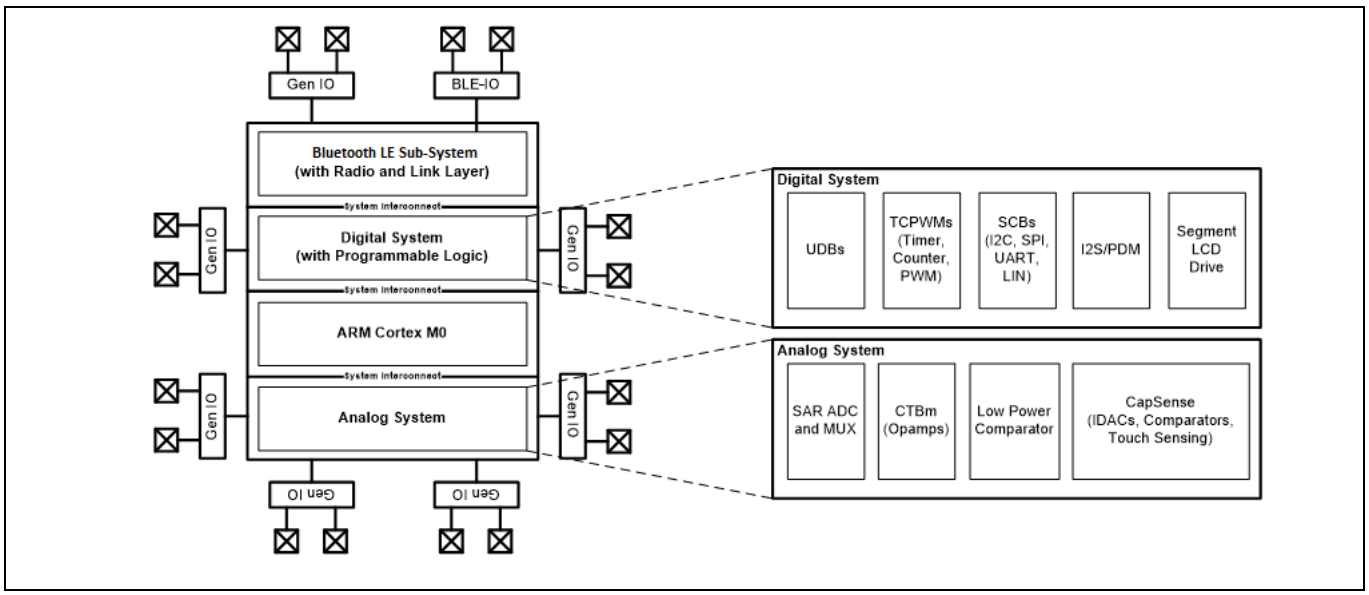


Figure 6 PSoC™ 4 with AIROC™ Bluetooth® LE 框图

典型的 MCU 要求 CPU 固件处理状态机、使用定时器来进行计时、并驱动一个输出引脚。因此，其功能路径几乎都是通过 CPU 实现的。但通过使用 PSoC™，可以进行异步并行操作。您可以配置 PSoC™中的组件，使其运行独立于 CPU。PSoC™ Creator 可在模拟、数字和可编程互联模块间实现无缝连接，并且在 PSoC™

PSoC™的性能比 Bluetooth® LE MCU 的更优越

Creator 中所有的模拟和数字外设均显示为虚拟的 IC 形式 (称为“组件”) 。本应用笔记中的示例项目说明了 PSoC™ Creator 及其组件的使用方法。

PSoC™也具有可编程数字模块，称为通用数字模块 (UDB) 。PSoC™ Creator 还提供了多个由 UDB 模块构成的组件，如：UART、SPI、I2S、定时器、PWM、计数器、数字门控 (如 AND、OR、NOT、XOR) 等。通过使用 PSoC™ Creator 中的 UDB，您也可以创建自己的自定义状态机和数字逻辑。欲了解该方法，请参考 **PSoC™ Creator 组件创建指南**。

5 Bluetooth® LE 概述

Bluetooth® LE 或 Bluetooth Smart™ 是一个由蓝牙技术联盟 (SIG) 定义的低功耗、短距离、低数据速率的无线通信协议。如 **Figure 7** 所示，Bluetooth® LE 具有一个分层协议栈，它可以在消耗小功率的条件下有效传输较小的数据量，该特性使其成为电池供电器件的首选无线协议。

Bluetooth® LE 堆栈的组成部分如下：

- 2.4 GHz RF 物理层 (PHY)，数据速率为 1 Mbps
- 链路层 (LL)，用于定义 PHY 的时序和数据包格式信息
- 主机控制接口 (HCI)，用于连接该堆栈中的硬件控制器 (PHY + LL) 层和固件主机层
- 逻辑链路控制和适配协议 (L2CAP)，可作为数据包装配/拆开和协议复用层使用
- 属性协议 (ATT) 定义了如何组织和访问应用数据
- 安全管理器 (SM)，提供了一个工具箱，用于保护 Bluetooth® LE 链路上的数据交换
- 通用属性配置文件 (GATT)，定义了访问由 ATT 层定义的数据的方式
- 通用访问配置文件 (GAP)，提供了一个应用定向接口，用于确定设备作为 Bluetooth® LE 连接的主设备还是从设备，并按照确定结果配置基础层

请参考**附录 E：Bluetooth® LE 协议**，了解 Bluetooth® LE 协议的详细说明。

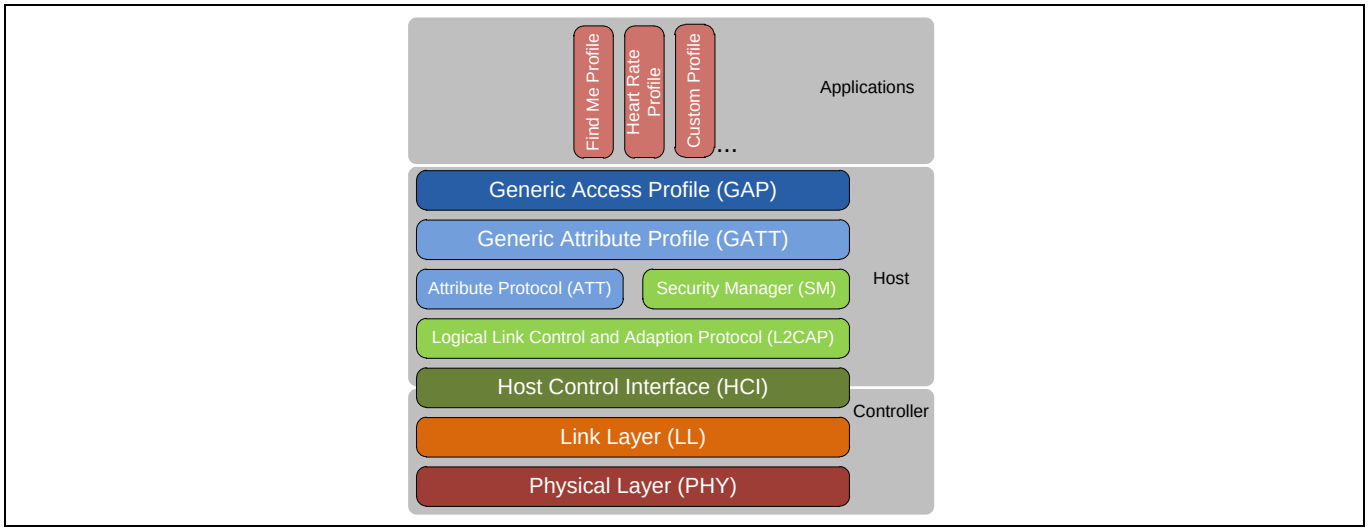


Figure 7 Bluetooth® LE 协议栈

若想开发 Bluetooth® LE 应用，便不需要了解 Bluetooth® LE 复杂协议栈的运作方式。Infineon 在 PSoC™ Creator 中提供了一个易于配置且基于 GUI 的 Bluetooth® LE 组件，这样有助于降低协议的复杂性。使用 Bluetooth® LE 前，应该掌握以下内容：

- Bluetooth® LE 连接的建立程序
- 应用数据表示和抽象化
- Bluetooth® LE GAP 和 GATT 层的配置与应用要求的匹配。

5.1 Bluetooth® LE 连接的建立程序

要想在两个设备 (例如：Infineon 的 Bluetooth® LE Pioneer 套件和一个智能手机) 之间建立 Bluetooth® LE 连接，需要掌握通用访问配置文件 (GAP) 的两种设备功能，如 **Figure 8** 所示：

- **GAP 外设：**该设备将其存在信息通知给 GAP 中心设备，并接受来自 GAP 中心设备的连接请求。实现心率测量功能的 Bluetooth® LE Pioneer 套件便是 GAP 外设设备的一个示例。
- GAP 中心设备：该设备将扫描 GAP 外设的存在通告并与它们进行连接。与心率测量设备连接的智能手机便是一个 GAP 中心设备示例。

中心设备与外设建立连接后，可认为两者之间已经通过 Bluetooth® LE 进行连接。在已建立的 Bluetooth® LE 连接上，通用属性配置文件 (GATT) 根据数据源和目标定义两个配置文件的角色 (与 GAP 的角色相互独立)，如 **Figure 8** 所示：

- **GATT 服务器：**GATT 服务器是一个包含了数据或状态的设备。通过一个 GATT 客户端配置时，它会将数据发送到 GATT 客户端或修改其局部状态。例如，心率测量设备是一个 GATT 服务器，它将心率数据发送给一个智能手机 GATT 客户端。同样，智能灯泡也是一个 GATT 服务器，它所具有的状态 (打开/关闭) 可通过智能开关 GATT 客户端得到配置。
- **GATT 客户端：**GATT 客户端可配置 GATT 服务器的状态或接收来自 GATT 服务器的数据。例如，从心率设备接收心率信息的智能手机即是一个客户端。

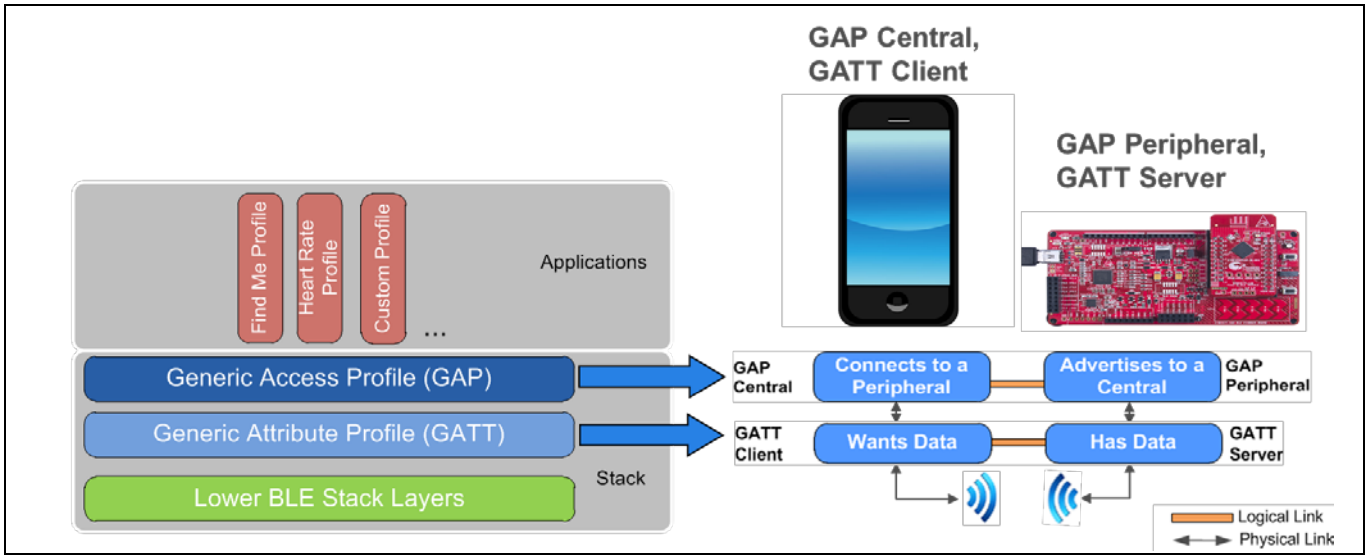


Figure 8 Bluetooth® LE 应用概述

建立 Bluetooth® LE 连接后，GATT 客户端将检测 GATT 服务器所保存的所有数据和状态。一旦完成检测，GATT 客户端可以对 GATT 服务器中的数据和状态进行配置和/或读/写操作。

5.2 GATT 数据格式

我们已经了解 Bluetooth® LE GAP/GATT 的功能，下面要了解在 GATT 服务器中如何存储数据，以及如何通过 GATT 客户端提取数据。GATT 服务器通过使用属性、特性和服务来表示和抽象化 Bluetooth® LE 器件中的数据。从本节中，您会看到：一个服务由一个或多个特性构成，而每个特性则由多个含有实际数据的属性构成。

- **属性：**一个属性包含 GATT 层的基础数据，它表示一条离散信息。如 **Figure 9** 所示，属性结构包括：
  - 属性句柄：用于寻址属性
  - 属性类型：一个由蓝牙 SIG 分配的 16 位通用唯一标识符 (UUID)，用于指定属性中所包含的数据
  - 属性值：包含实际数据
  - 属性权限：指定读/写权限和属性的安全性要求

心率测量、电池电量、电池电量单位及器件名称均为属性示例。

GATT 服务器包含多项属性，它们被存储在名称为“属性数据库”的数据库中。通过属性句柄，GATT 客户端可对 GATT 服务器的属性数据库中的一个或多个属性执行读/写操作。各位用户仅要知道您要执行读/写的属性句柄，而其他信息则由 Infineon 的 Bluetooth® LE 组件 API 函数进行抽象化。

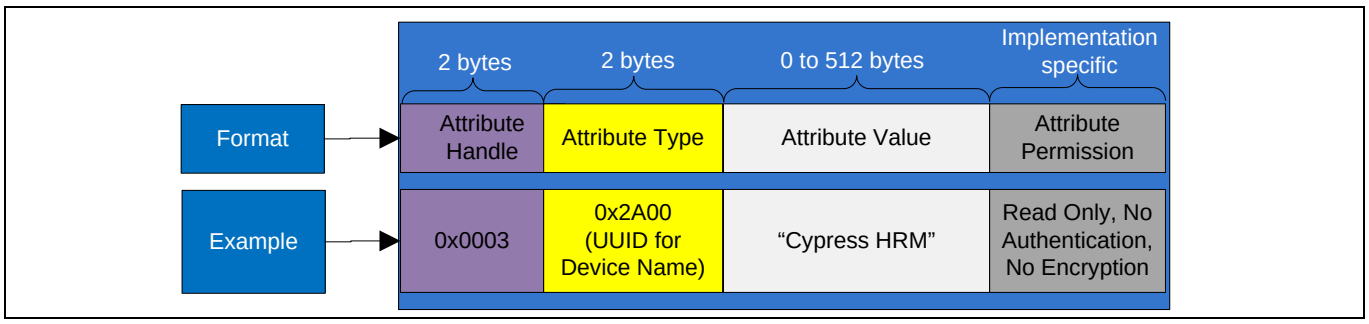


Figure 9 GATT 属性示例

- **特性：**一个特性由多个离散属性组成，这些属性结合在一起，可定义系统信息或有意义的数据。每项特性均包含一个特性声明属性、一个特性值属性以及一个或多个可选的特性描述符属性，如 **Figure 10** 所示。例如，将电池电量属性“90”和电池电量描述符属性“%”结合在一起时，则表示系统的电池电量为 90%的信息。

蓝牙 SIG 提供了一套预定义的标准特性，您可以将标准特性使用到应用中，或者创建自己的自定义特性。智能灯泡的打开/关闭状态即是一个自定义特性。

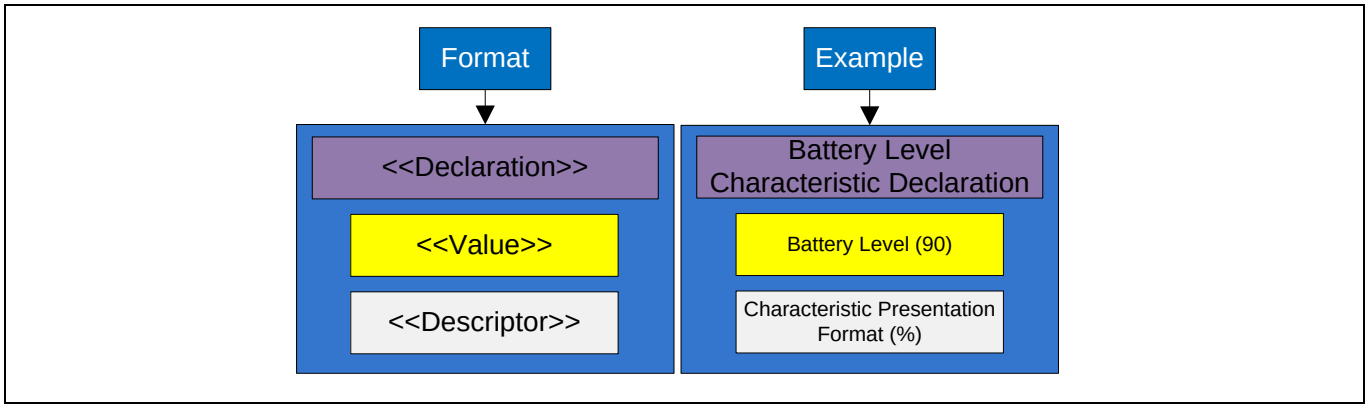


Figure 10 GATT 特性示例

- **服务：**一个服务由一个或多个相关特性组成，用于定义器件的某个特殊功能或性质。**Figure 11** 显示了一个心率服务示例，它具有的三个特性描述了与心率测量相关的信息。

蓝牙 SIG 还提供了一套预定义的标准服务，用于实现通用的 Bluetooth® LE 器件功能。另外，您可以创建自己的自定义服务，其中可能包含标准或自定义特性。智能灯泡即是一个自定义服务，包含了灯泡的打开/关闭状态特性。

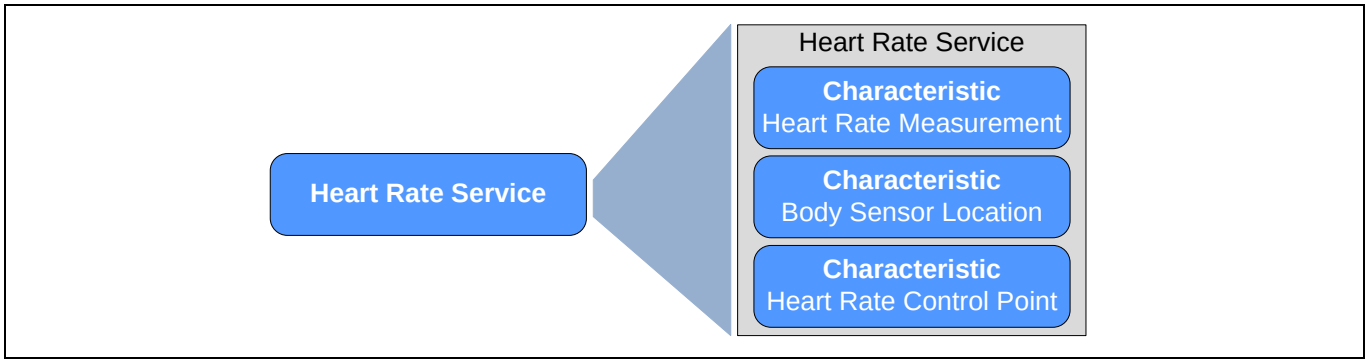


Figure 11　GATT 服务示例

除了定义数据格式以外，GATT 层还定义了一套程序或方法，用于检测和访问 GATT 层的数据。建立 Bluetooth® LE 连接后，GATT 客户端将使用这些 GATT 程序检测所连接 GATT 服务器的整个属性数据库。完成该操作后，GATT 客户端将通过属性句柄使用特性读/写 GATT 程序对属性执行读/写操作。

Figure 12 显示了心率数据在心率传感器上显示的情况，以及智能手机通过 GATT 程序对心率数据进行的提取操作。典型的心率传感器执行了 GAP 外设角色，因为它将自己的存在信息通知给一个 GAP 中心设备 (如智能手机) 并与其进行连接。连接建立后，心率传感器成为一个 GATT 服务器，并且它保存了心率数据。GATT 服务器支持包含心率测量特性的心率服务，心率测量特性又存储了具有实际心率测量值的心率值属性。

而智能手机 GATT 客户端在建立连接后会通过 GATT 检测程序初步检测所连接 GATT 服务器支持的所有服务、特性和属性。完成该操作后，智能手机 GATT 客户端将使用 GATT 特性程序从 GATT 服务器接收心率数据。

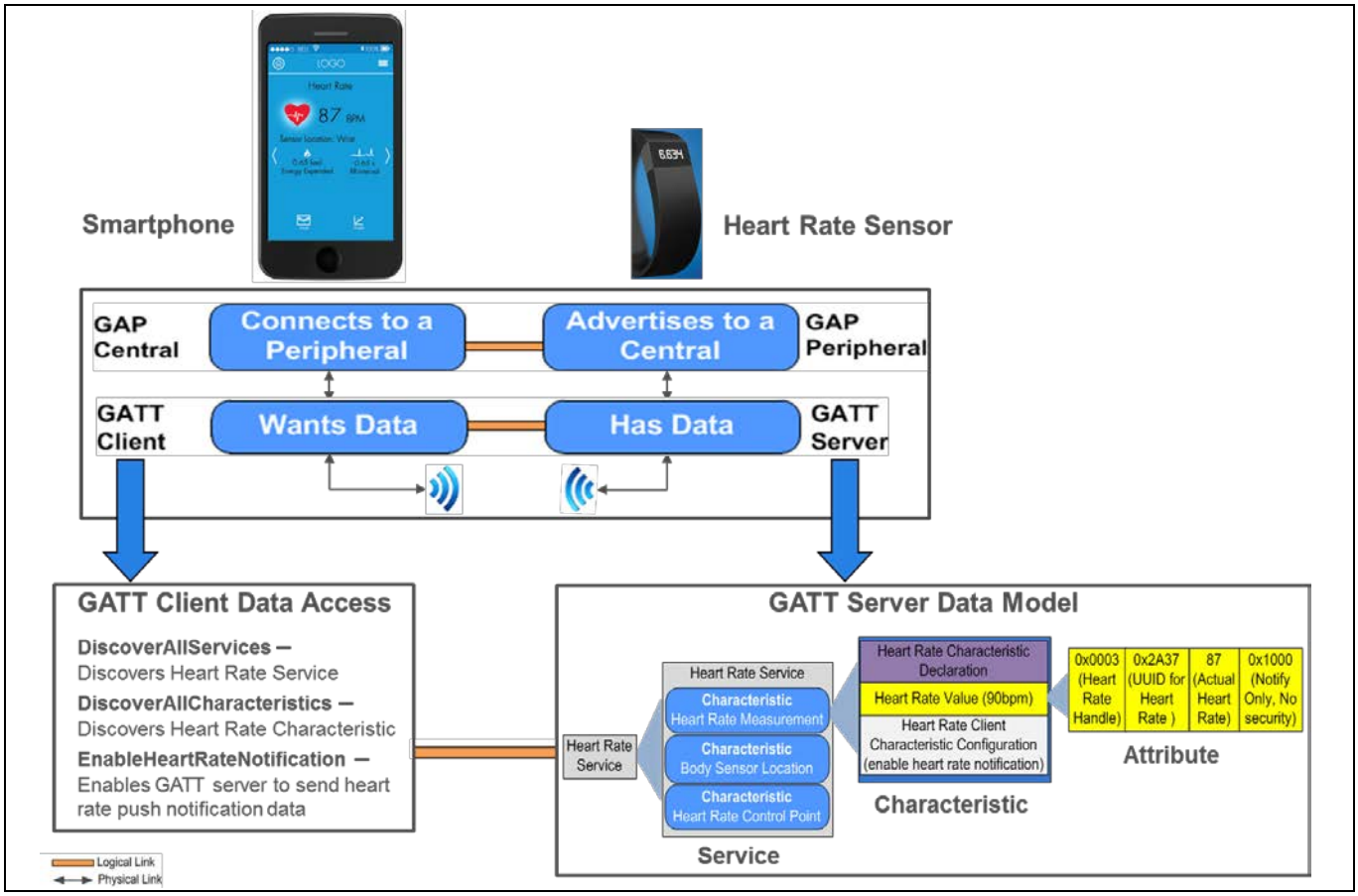


Figure 12 GATT 数据格式

5.3 Bluetooth® LE 配置文件

Bluetooth® LE 配置文件是一种蓝牙规范，它保证配置文件互相兼容的各个设备间应用级的相互操作性。它定义了受支持的不同 Bluetooth® LE 层的功能和配置以及 GATT 服务，从而可创建一个专用终端应用或使用情况。例如，如果您想实现一个心率监控设备，那么 Bluetooth® LE 心率配置文件定义需要支持的 GAP、GATT 功能以及 GATT 服务，从而创建一个可相互操作的心率监控设备。蓝牙 SIG 提供了一套预定义的标准配置文件，用于通用的 Bluetooth® LE 终端应用。另外，您可以创建自己的自定义配置文件，其中可包含标准或自定义服务。

如 **Figure 13** 所示，与 GATT 层一样，配置文件定义了两个应用功能：

- **传感器或服务器：**传感器配置文件功能由含有数据的应用所支持。传感器配置文件规范定义了 Bluetooth® LE 器件的所需功能 (如 GATT/GAP 功能) 和行为 (如通告间隔、所支持的 GATT 服务) ，以支持传感器应用的使用情况。遵循传感器配置文件可保证传感器应用可与执行相应收集器配置文件的所有器件相互操作。例如，执行心率传感器配置文件的心率监控设备可与执行心率收集器配置文件的所有智能手机相互操作。
- **收集器或客户端：**收集器配置文件角色由需要提取数据的应用承担。收集器配置文件规范定义了所需的 Bluetooth® LE 器件功能和行为，用于对任何执行相应传感器配置文件规范的器件进行相互操作和提取信息。

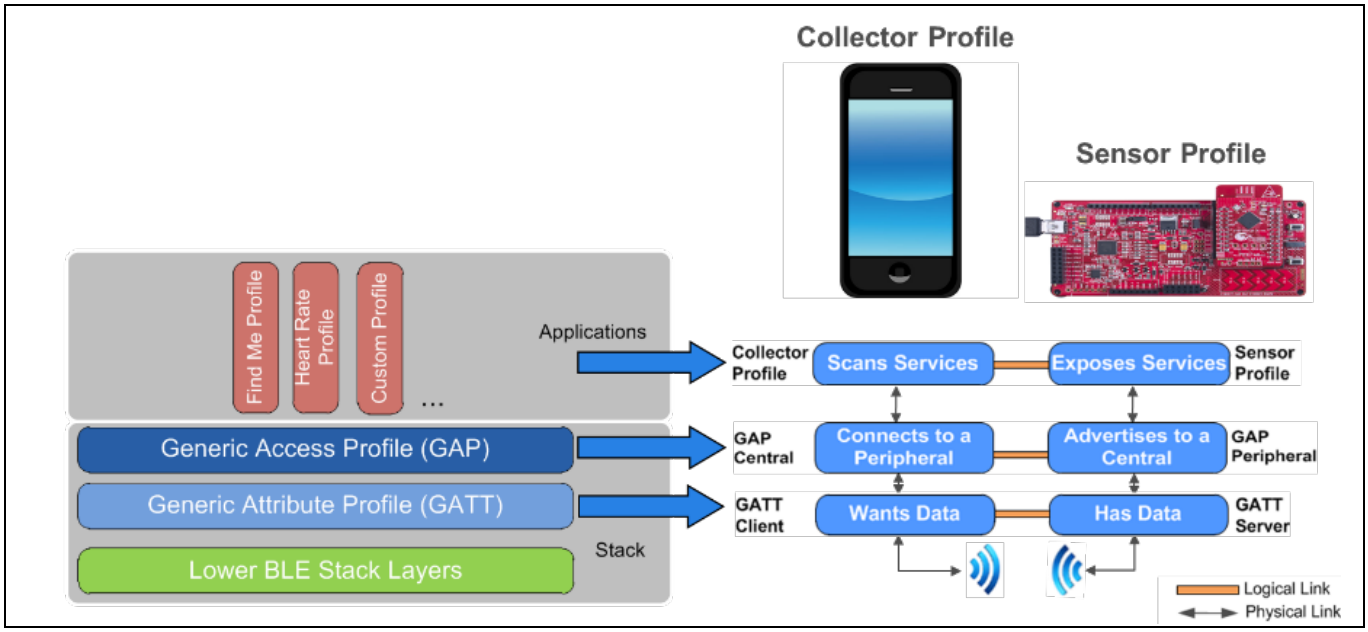


Figure 13 Bluetooth® LE 配置文件示例

Figure 14 总结了 Bluetooth® LE 器件中的数据抽象化和层次结构。

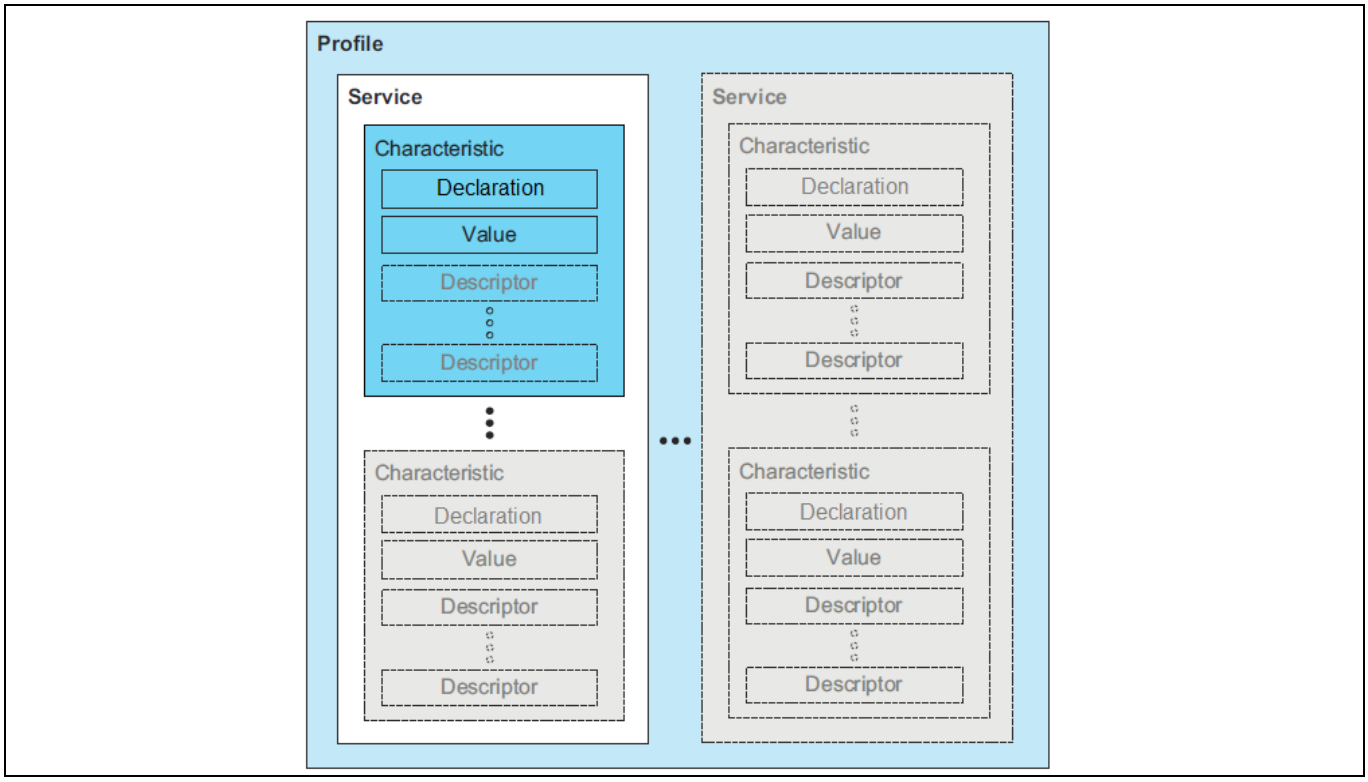


Figure 14 Bluetooth® LE 数据的层次结构*

*图片由蓝牙 SIG 提供

5.4 Bluetooth® LE 组件

5.4.1 特性

PSoC™ Creator 中的 Bluetooth® LE 组件将 Bluetooth® LE 协议抽象化为一个简单易用的 GUI 和一个具有少数函数的 API。Bluetooth® LE 组件能够：

- 支持符合蓝牙版本 4.2 的协议栈
- 支持所有 GAP 角色 – 中心设备、外设、广播者和观察者。另外，还支持一些 GAP 角色的结合，如中心设备和观察者、外设和广播者。
- 支持由蓝牙 SIG 定义的所有 GATT 配置文件和服务，并且为每一个受支持的配置文件提供至少一个示例代码
- 通过 Bluetooth® LE 组件 GUI 简化了自定义配置文件的创建和使用
- 支持 L2CAP 面向连接通道、链路层低占空比广告和 LE Ping 等特性
- 支持能保存最多 4 个绑定器件的列表和 8 个器件的白名单
- 支持所有蓝牙 4.2 安全模式
- 支持蓝牙 4.2 规范的链路层数据长度扩展、链路层保密以及 LE 安全连接特性

有关详细信息，请参见 Bluetooth® LE 组件**数据手册**。

5.4.2 配置

使用 Bluetooth® LE 组件时，需要执行 5 个步骤来创建一个基于蓝牙 SIG 定义的配置文件 (标准配置文件) 应用。例如，若想创建**我的第一个 PSoC™ 4 with AIROC™ Bluetooth® LE 设计**一节中所描述的 Find Me Target 示例，请遵循以下步骤：

步骤 1：选择设计所需的配置文件和配置文件功能。Bluetooth® LE 组件将自动为选定的配置文件功能选择所需 GAP 角色和 GATT 角色。对于实现目标角色的由蓝牙 SIG 定义的 Find Me 配置文件 (如 **Figure 15**)，GAP 角色和 GATT 角色根据 **Find Me 配置文件**规范分别被设置为 Peripheral 和 Server。

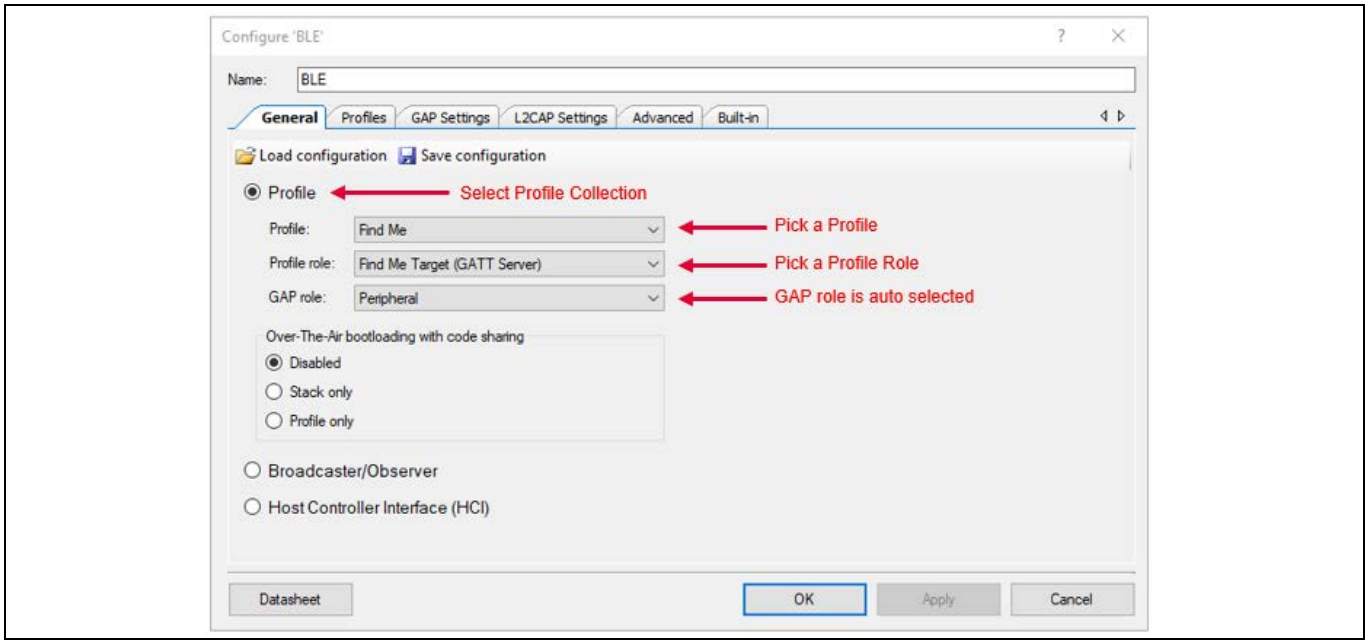


Figure 15 Bluetooth® LE 组件配置文件的配置

步骤 2：对于所选的配置文件功能，将根据配置文件规范自动生成所有受支持的 GATT 服务和相应特性。需要时，可以根据设计进行验证和/或编辑各个服务和特性值。**Profiles** 选项卡中的各种配置构成了所选设计的属性数据库。

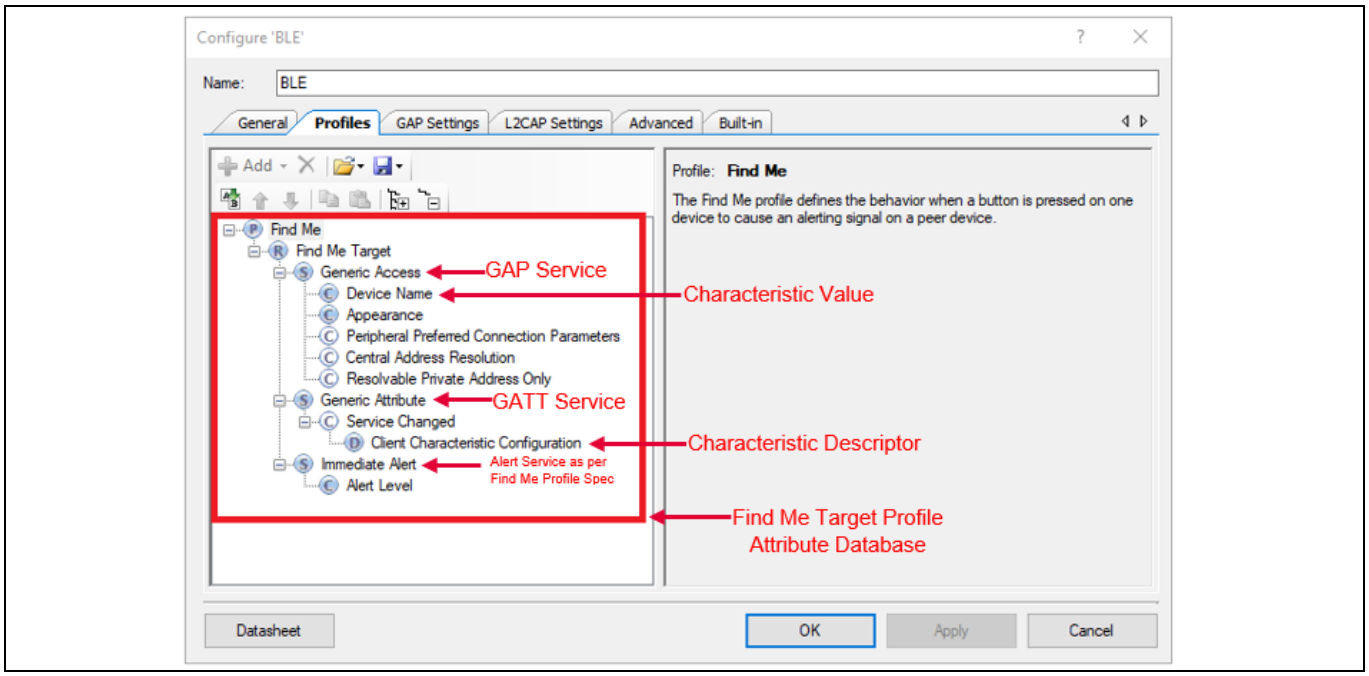


Figure 16 Bluetooth® LE 属性数据库配置

步骤 3：为您的设计配置 GAP 通用和广告数据设置 (第 14 页中最上面部分已提供了解释)，如 **Figure 17** 和 **Figure 18** 所示。

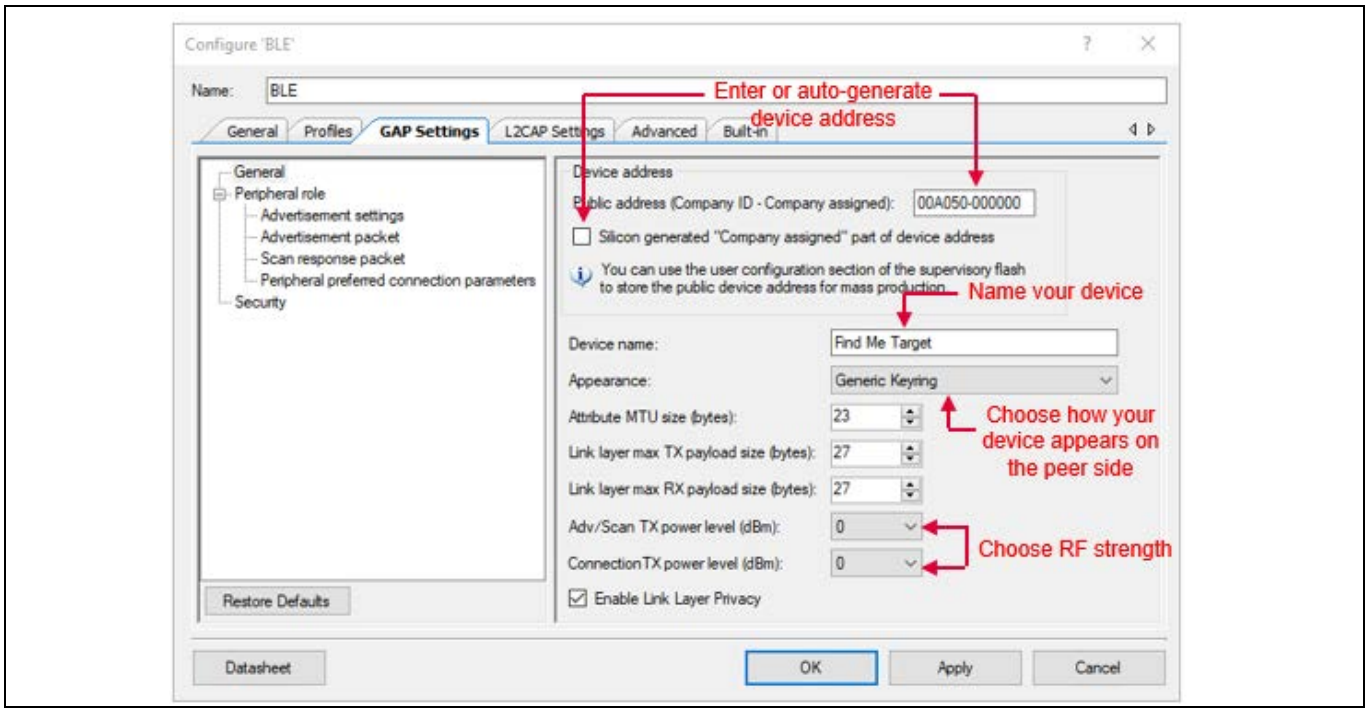


Figure 17 Bluetooth® LE GAP 配置

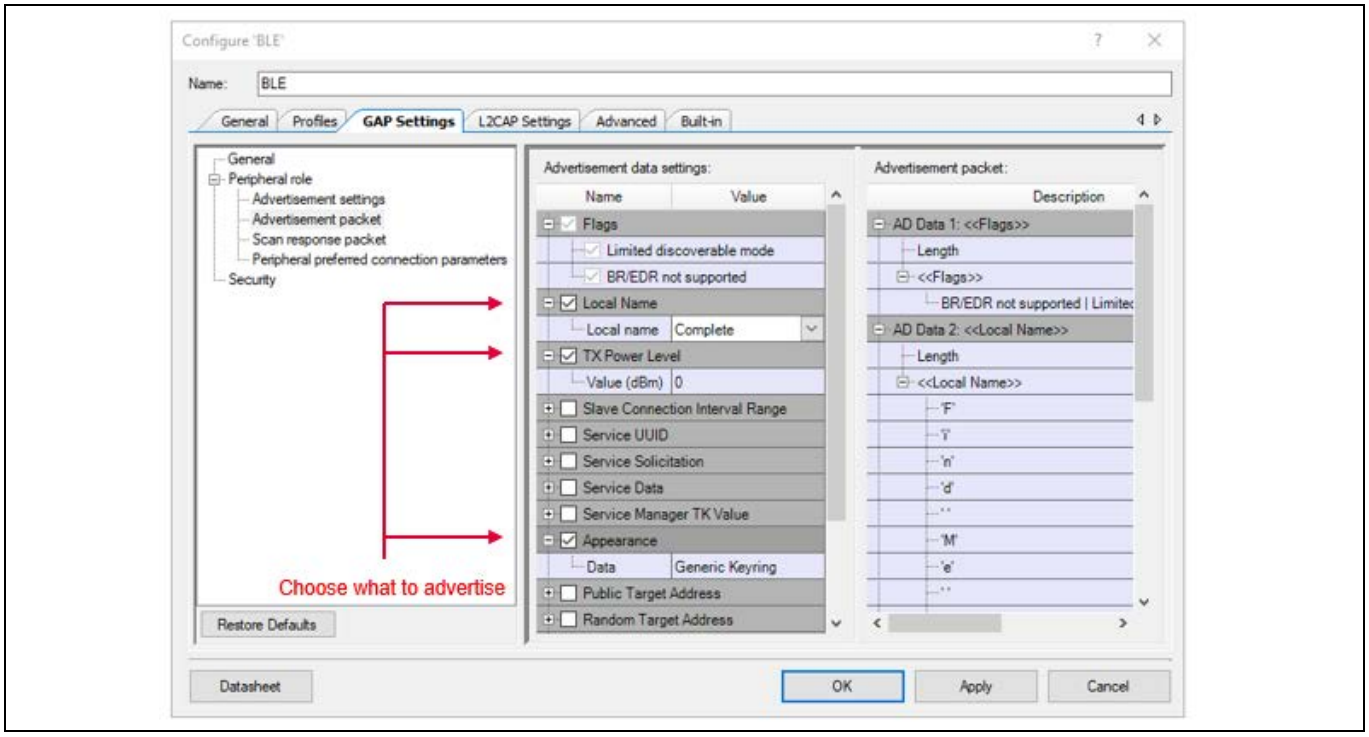


Figure 18 Bluetooth® LE 广告数据配置

步骤 4：编写固件，以初始化您刚配置好的设计。为 Bluetooth® LE 组件注册寄存器事件处理程序，从而能够接收事件和数据。本文档的后续章节中对事件处理程序进行了详细介绍。

步骤 5：在程序主循环中等待来自 Bluetooth® LE 组件的事件，并通过使用 Bluetooth® LE 组件 API 函数进行所需操作或将数据发送给中心设备。

我的第一个 PSoC™ 4 with AIROC™ Bluetooth® LE 设计一节有助于您通过 Bluetooth® LE 组件的每一个配置步骤创建一个简单的外设应用。请参考应用笔记 **AN91184** 和 **AN91162**，了解如何使用 Bluetooth® LE 组件和 Bluetooth® LE 标准以及自定义配置文件进行开发应用的详细说明。

6 PSoC™ 4 with AIROC™ Bluetooth® LE 开发设置

Figure 19 显示了通过 PSoC™ 4 with AIROC™ Bluetooth® LE 器件评估各种 Bluetooth® LE 外设设计所需的硬件和软件工具。Bluetooth® LE Pioneer 套件 (**Figure 19** 中的红色电路板) 通常被配置为一个外设，与 CySmart iOS/Android 应用或 CySmart 主机仿真工具等中心设备进行通信。CySmart 主机仿真工具还需要一个 Bluetooth® LE 收发器 (**Figure 19** 中的黑色电路板) 才能运行。

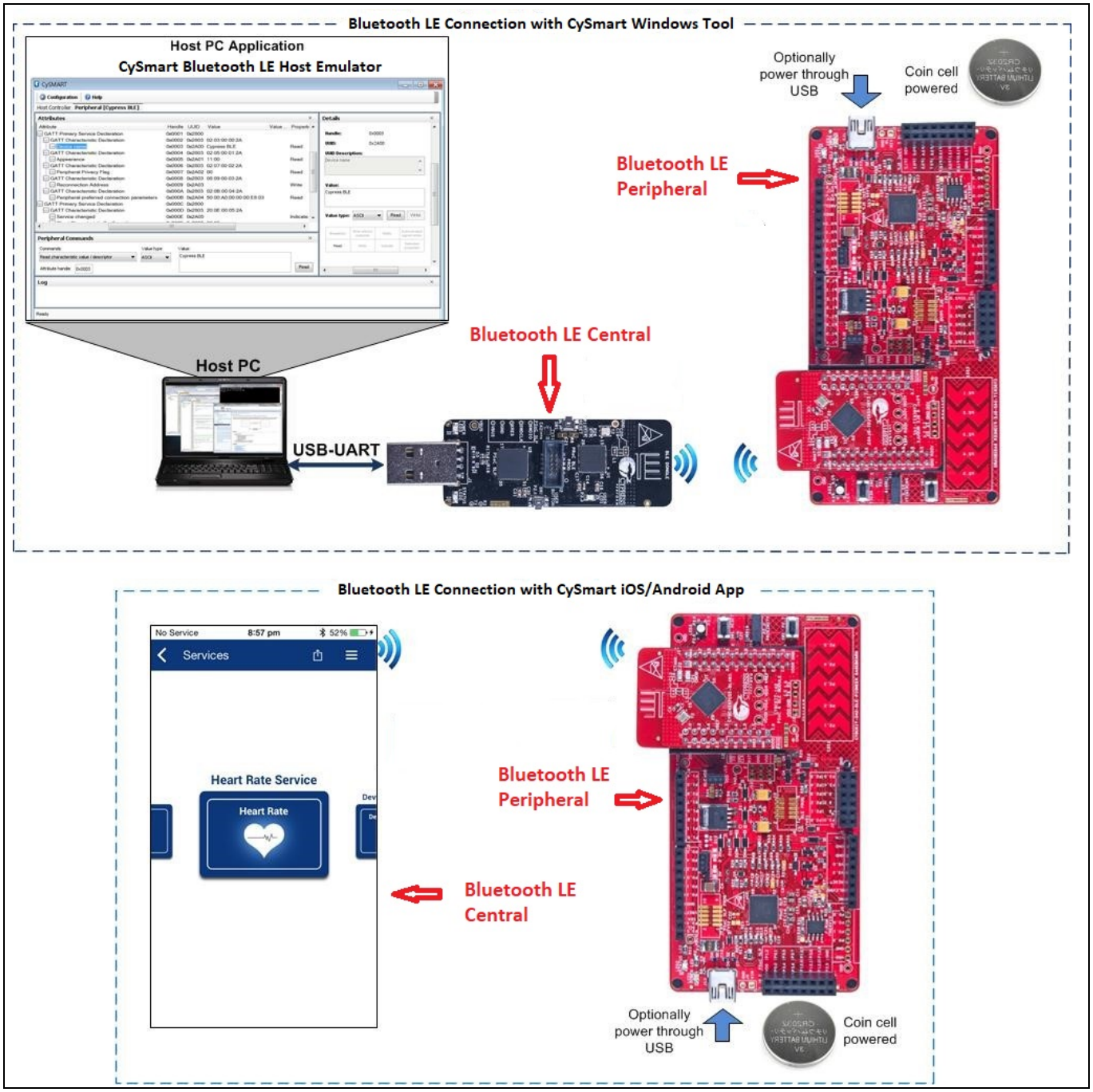


Figure 19 Bluetooth® LE 功能设置

如 **Figure 20** 所示，Bluetooth® LE 收发器经过预编程，从而可与 Windows CySmart 主机仿真工具配合使用。Bluetooth® LE Pioneer 套件具有一个板上 USB 编程器，该编程器与 PSoC™ Creator 结合使用时能够编程

PSoC™ 4 with AIROC™ Bluetooth® LE 开发设置

和调试您的 Bluetooth® LE 设计。Bluetooth® LE Pioneer 套件可由 USB 接口或纽扣电池供电。Bluetooth® LE 收发器和 Bluetooth® LE Pioneer 套件可同时连接到同一个主机 PC 上，以进行开发和测试。

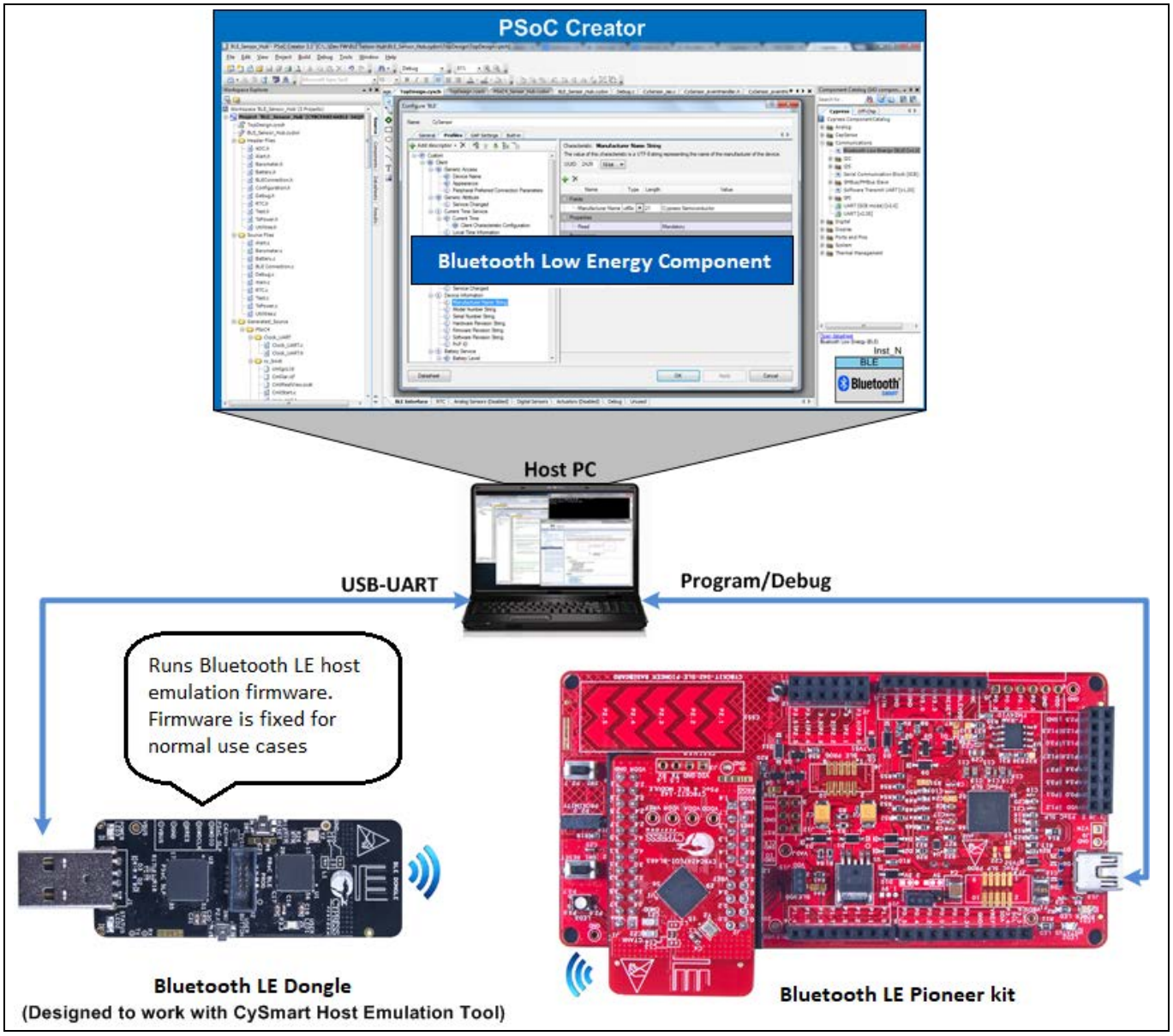


Figure 20 Bluetooth® LE 开发设置

7 我的第一个 PSoC™ 4 with AIROC™ Bluetooth® LE 设计

本节逐步介绍了通过 PSoC™ Creator 构建使用 PSoC™ 4 with AIROC™ Bluetooth® LE 器件的简单 Bluetooth® LE Pioneer 套件设计的流程。它描述了创建蓝牙 SIG 定义的简单标准配置文件设计的各个步骤。若想创建高标准或自定义配置文件设计，请参考**设计 Bluetooth® LE 应用**和**创建 Bluetooth® LE 自定义配置文件**应用笔记。

7.1 设计概述

本设计实现了担任目标角色的 Bluetooth® LE **Find Me 配置文件 (FMP)**，它包含了一个即时警报服务 (**IAS**)。FMP 和 IAS 分别是由蓝牙 SIG 定义的 Bluetooth® LE 标准配置文件和服务。Bluetooth® LE Pioneer 套件上的 LED 状态变化表示由 Find Me Locator 触发的警报级别，如 **Figure 21** 所示。这两个状态 LED 表示 Bluetooth® LE 接口的状态。

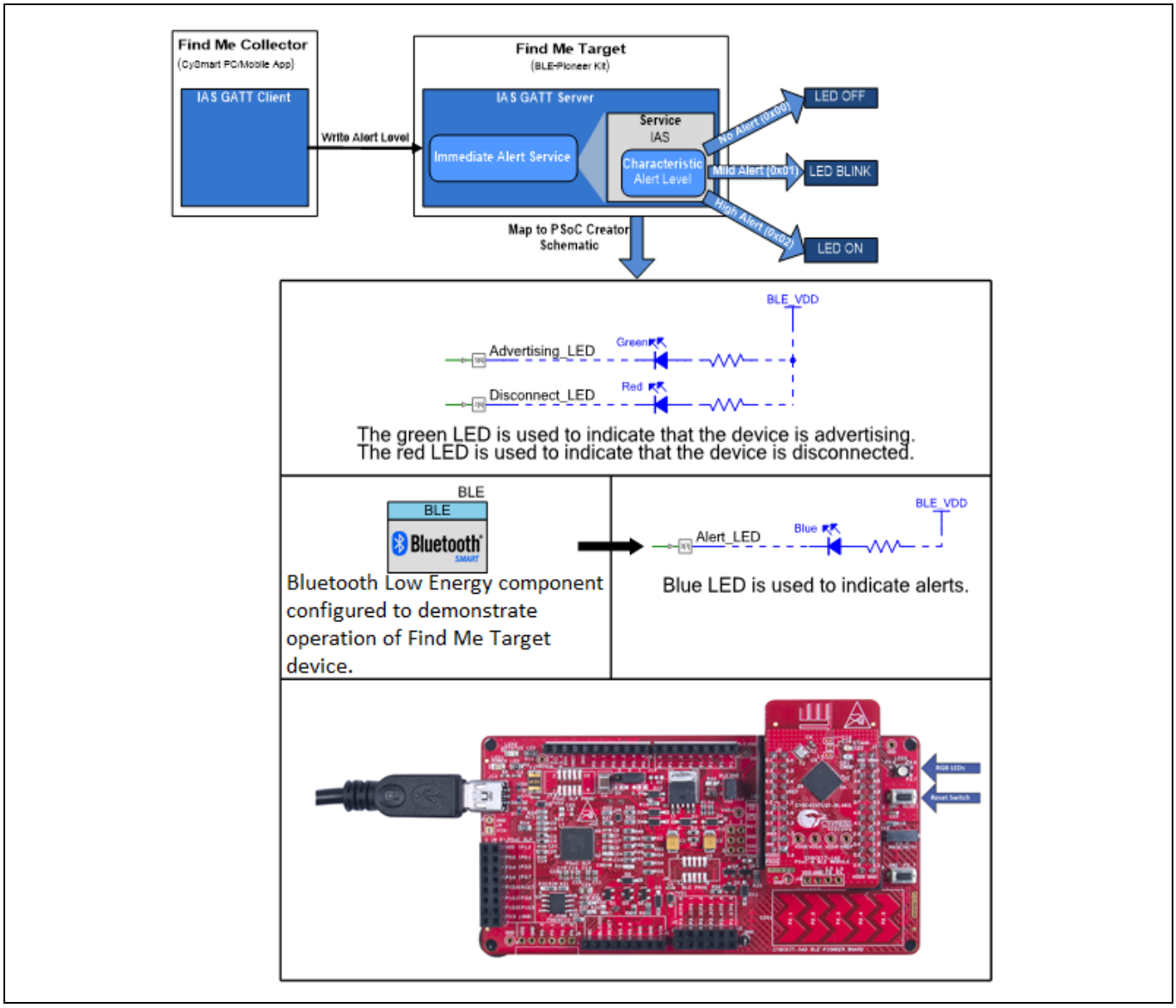


Figure 21 我的第一个 PSoC™ 4 with AIROC™ Bluetooth® LE 设计

我的第一个 PSoC™ 4 with AIROC™ Bluetooth® LE 设计

通过下面四个步骤，可以创建自己第一个 PSoC™ 4 with AIROC™ Bluetooth® LE 设计：

1. 在 PSoC™ Creator 原理图页面中构建设计。
2. 编写固件，以初始化和处理 Bluetooth® LE 事件。
3. 在 Bluetooth® LE Pioneer 套件上编程 PSoC™ 4 with AIROC™ Bluetooth® LE 器件。
4. 使用 CySmart 主机仿真工具或手机应用测试您的设计。

Note: *用于本应用笔记中描述的 Bluetooth® LE 示例设计的实用 PSoC™ Creator 项目是 PSoC™ Creator 代码示例的一部分。若想使用代码示例，请依次选择* ***File > Code Example****，然后将* ***Filter by*** *选为* ***Find Me> BLE_FindMe****。如果您使用了代码示例，请跳过前面介绍的原理图配置 (步骤 1) 和固件开发 (步骤 2)，并执行编程 (步骤 3) 和测试设计 (步骤 4)。*

7.2 先决条件

在开始构建设计前，请确保您已拥有一个 **Bluetooth® LE Pioneer 套件**，并安装了以下软件：

- **PSoC™ Creator 4.4**，内置 **PSoC™ Programmer 3.29**
- **CySmart 主机仿真工具**或 CySmart **iOS**/**Android** 应用

7.3 第一步：构建设计

本节逐步向您介绍了设计流程。首先是构建一个空白项目，然后指导您通过原理图进行设计。

1. 在您的 PC 上安装 PSoC™ Creator 4.4 或更高版本。
2. 启动 PSoC™ Creator，然后从 **File** 菜单中依次选择 **New** > **Project**，如 **Figure 22** 所示。

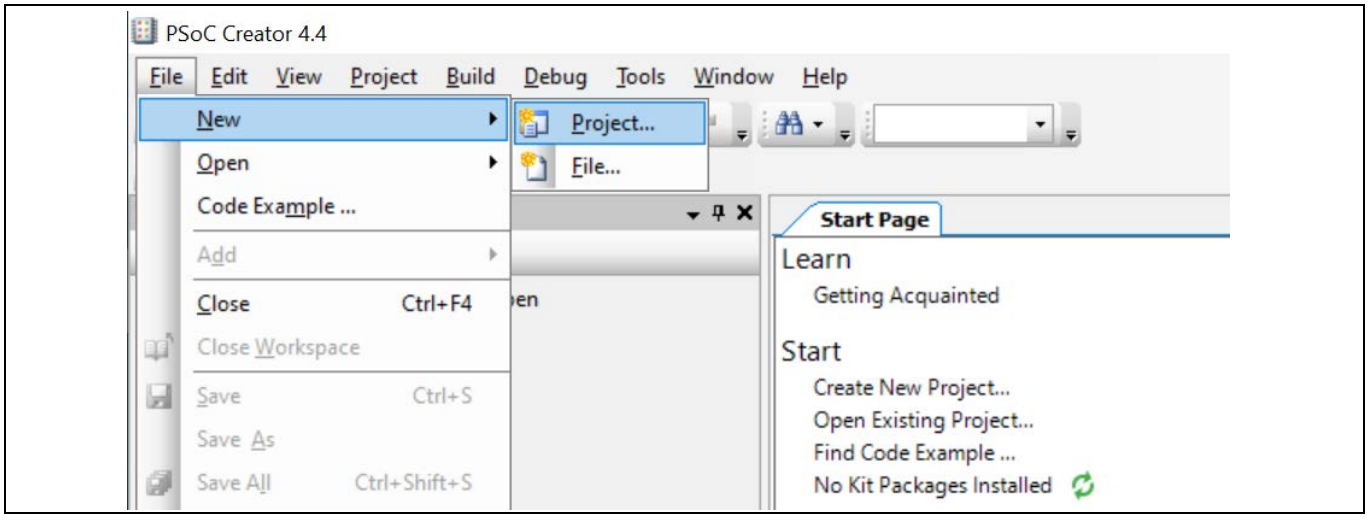


Figure 22 创建新项目

3. 将目标器件选择为“PSoC™ 4” (如 **Figure 23** 所示)，从而选择 **Bluetooth® LE Pioneer 套件**上使用的 CY8C4248LQI-BL583 器件，然后点击 **Next**。要选择 CY8C4248LQI-BL583 器件，请单击下拉列表并选择“Launch Device Selector”并选择提到的器件。如果您正在使用一个自定义 PSoC™ 4 with AIROC™ Bluetooth® LE 硬件或其他的 PSoC™ 4 with AIROC™ Bluetooth® LE 器件编号，请在目标器件上选择“Launch Device Selector”选项，然后选择相应的器件编号。

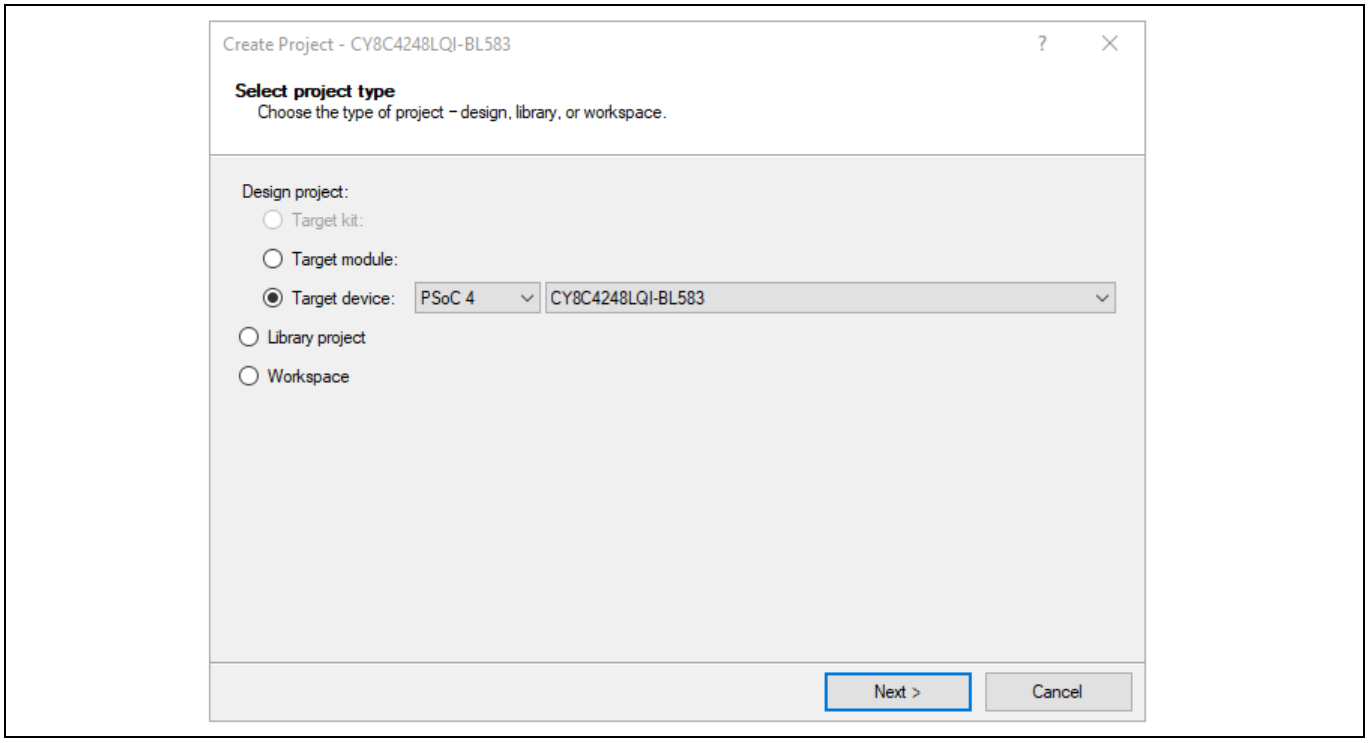


Figure 23　选择目标器件

4. 选择一个“Empty Schematic”项目模板，如 **Figure 24** 所示，然后点击 **Next**。

Note: *如果您想使用预建示例，请选择 **Figure 24** 中的“Code example”，并点击 **Next**。在代码示例选择窗口中，将 **Filter by** 选为 **Find Me> BLE_FindMe**，并直接跳到编程章节。*

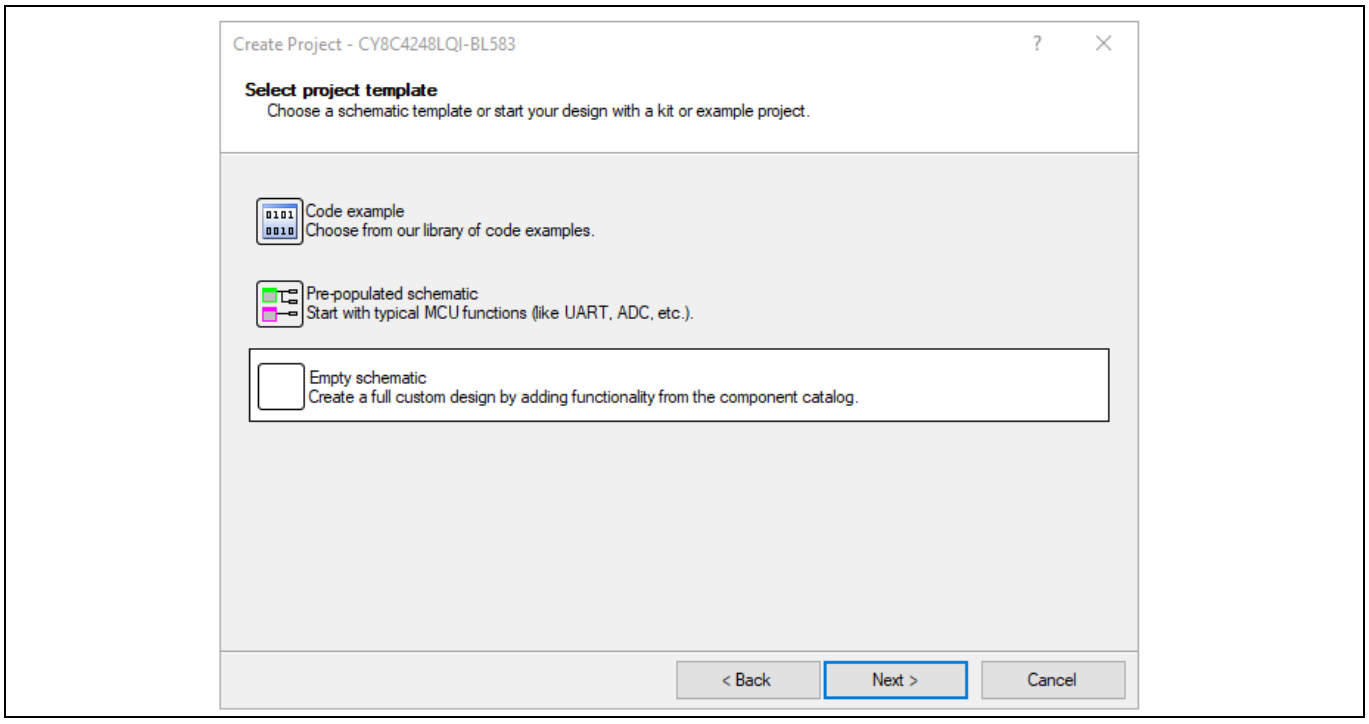


Figure 24　选择项目模板

5. 给工作区和项目命名，如“BLE_FindMe”，如 **Figure 25** 所示。为新项目选择相应的位置，然后点击 **Finish**。

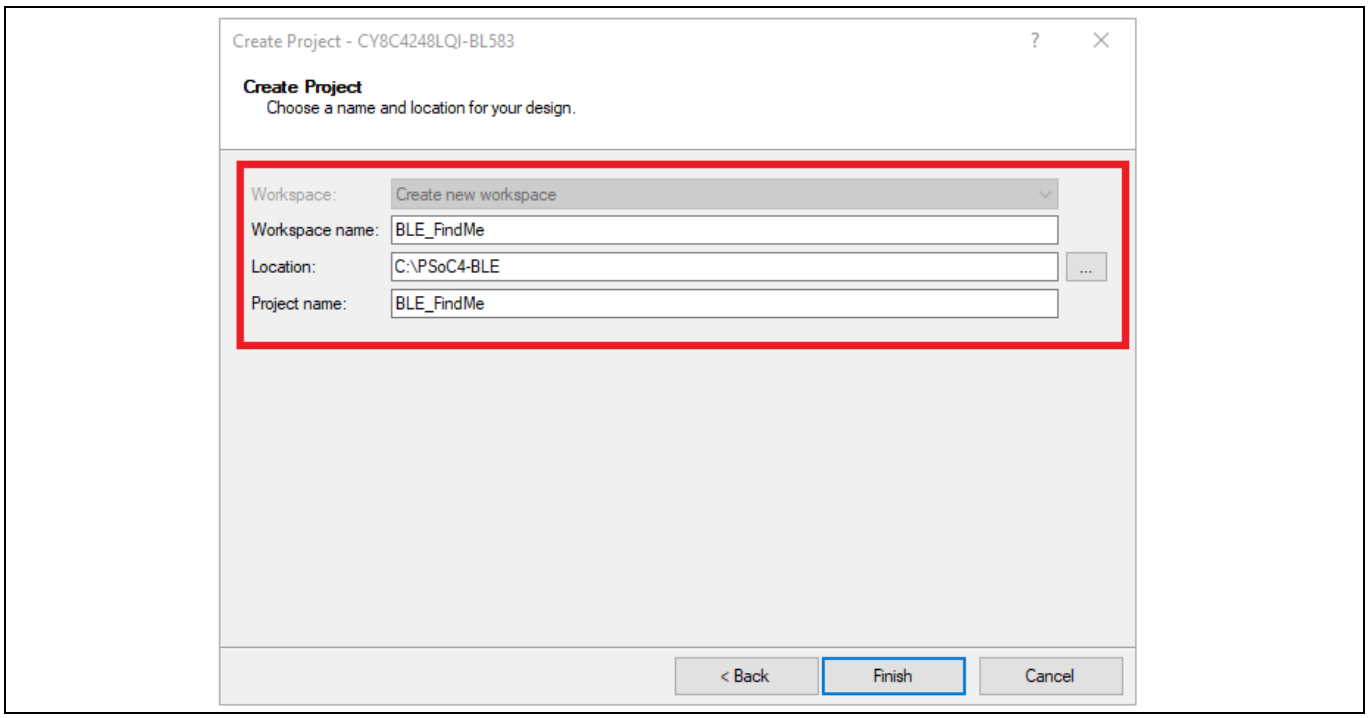


Figure 25　为项目命名

创建新项目会生成含有基本文件的项目文件夹。您可以在 **Workspace Explorer** 窗口查看这些文件，如 **Figure 26** 所示。

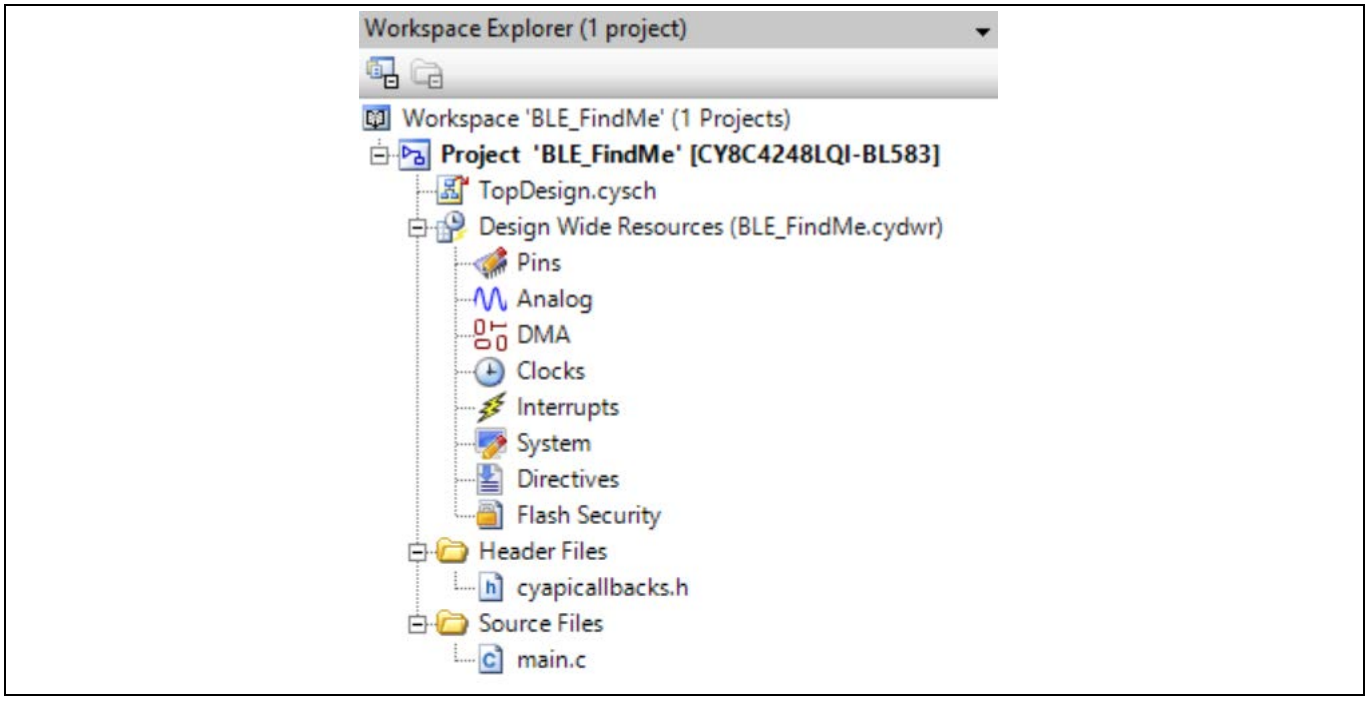


Figure 26　打开 TopDesign 原理图

6. 双击项目原理图文件 *TopDesign.cysch*，即可打开它。
7. 在项目原理图中，您可以通过拖放组件并连接它们来创建模拟、数字和通信设计。对于当前设计，您需要使用两个组件：Bluetooth® LE 和引脚 — 如 **Figure 21** 所示。
8. 将 **Component Catalog** 中的某个 Bluetooth® Low Energy 组件拖放到原理图中，如 **Figure 27** 所示。

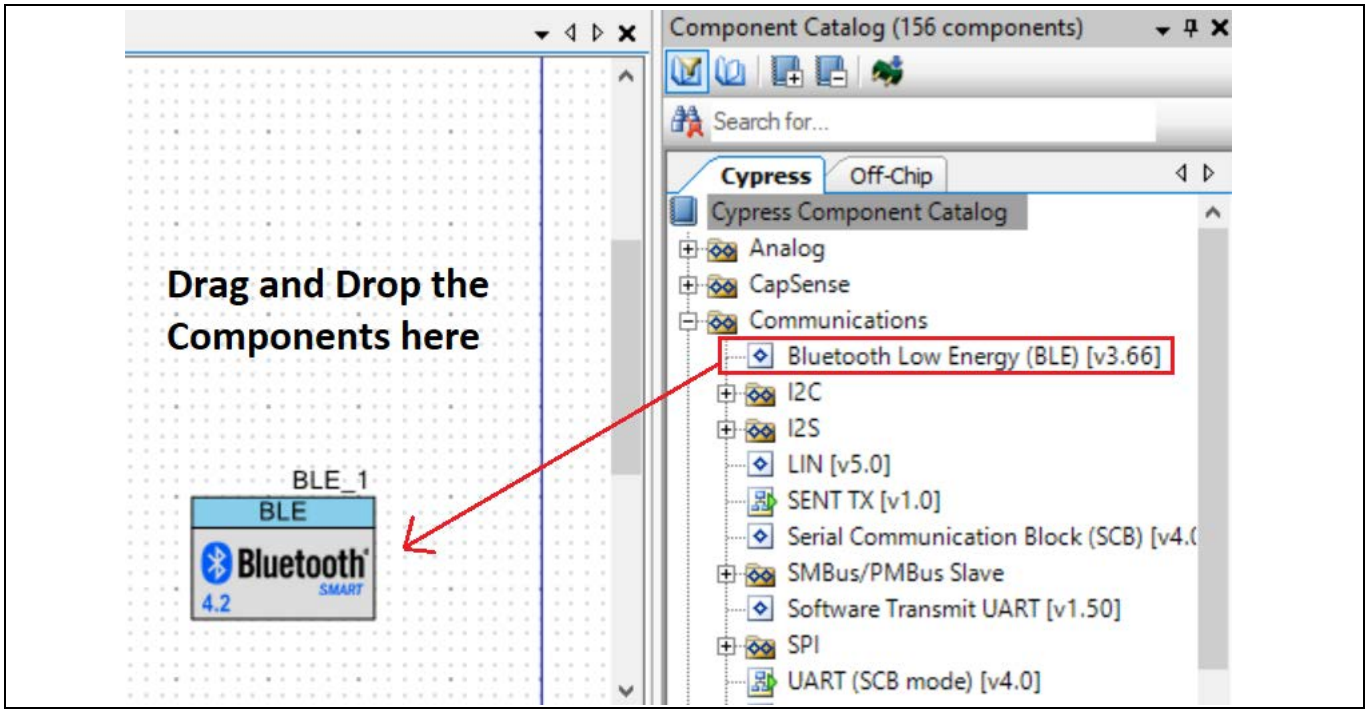


Figure 27 Bluetooth® LE 组件的位置

9. 双击原理图中的 Bluetooth® LE 组件，然后将其配置为带有以下属性的“BLE Find Me Target”：

- GAP 外设功能设为 Find Me Target (GATT Server)，如 **Figure 28** 所示。
- GAP Device Name 设为“Find Me Target”，Appearance 设为“Generic Keyring”，然后勾选“Silicon generated “Company assigned” part of device address”项，如 **Figure 29** 所示。这样便配置了其他器件开发您的器件时器件所出现的名称和类型，并为您的器件分配了一个唯一的 Bluetooth® LE 器件地址。
- 受限制的广告模式，其中广告超时为 30 秒，快速广告间隔为 20 到 30 ms，如 **Figure 30** 所示。快速广告允许快速开发和连接，但这样会增加 RF 广告数据包，从而消耗更多功耗。取消选择“Slow advertising interval”复选框。
- 勾选 Advertisement Packet 的 Immediate Alert Service (如 **Figure 31** 所示) 以及 Scan Response Packet 的 Local Name、Tx Power Level 和 Appearance 等字段 (如 **Figure 32** 所示) 。
- GAP 安全性被设置为最低等级，即数据转换时不需要认证、加密或授权 (模式 1，无安全性) 。将 I/O 功能和绑定要求分别设置为“No Input No Output”和“No Bonding”。

Figure 28 到 **Figure 34** 分别显示了该配置的 Bluetooth® LE 组件屏幕截图。

Note: *对于该设计，您不需要更改 GAP Settings > Peripheral preferred connection parameters、Profiles、L2CAP Settings 和 Advanced 选项卡中 Bluetooth® LE 组件的默认配置内容。*

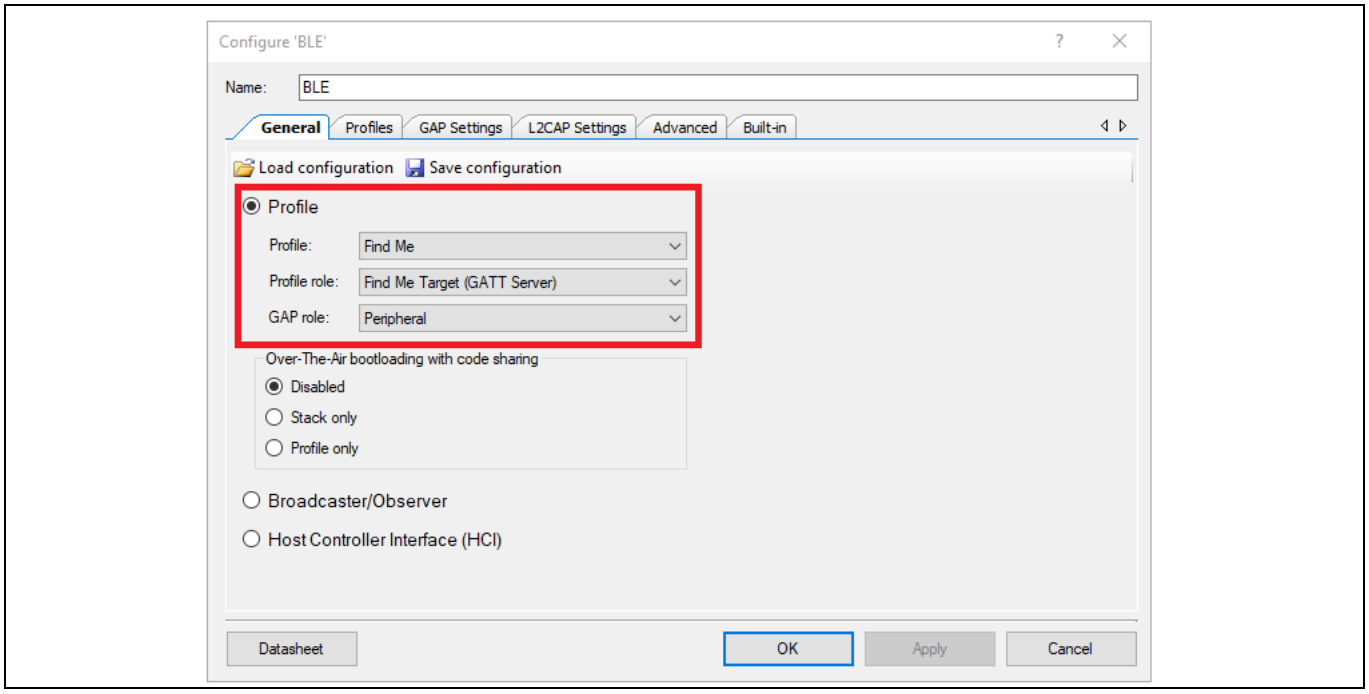


Figure 28 Bluetooth® LE 组件通用配置

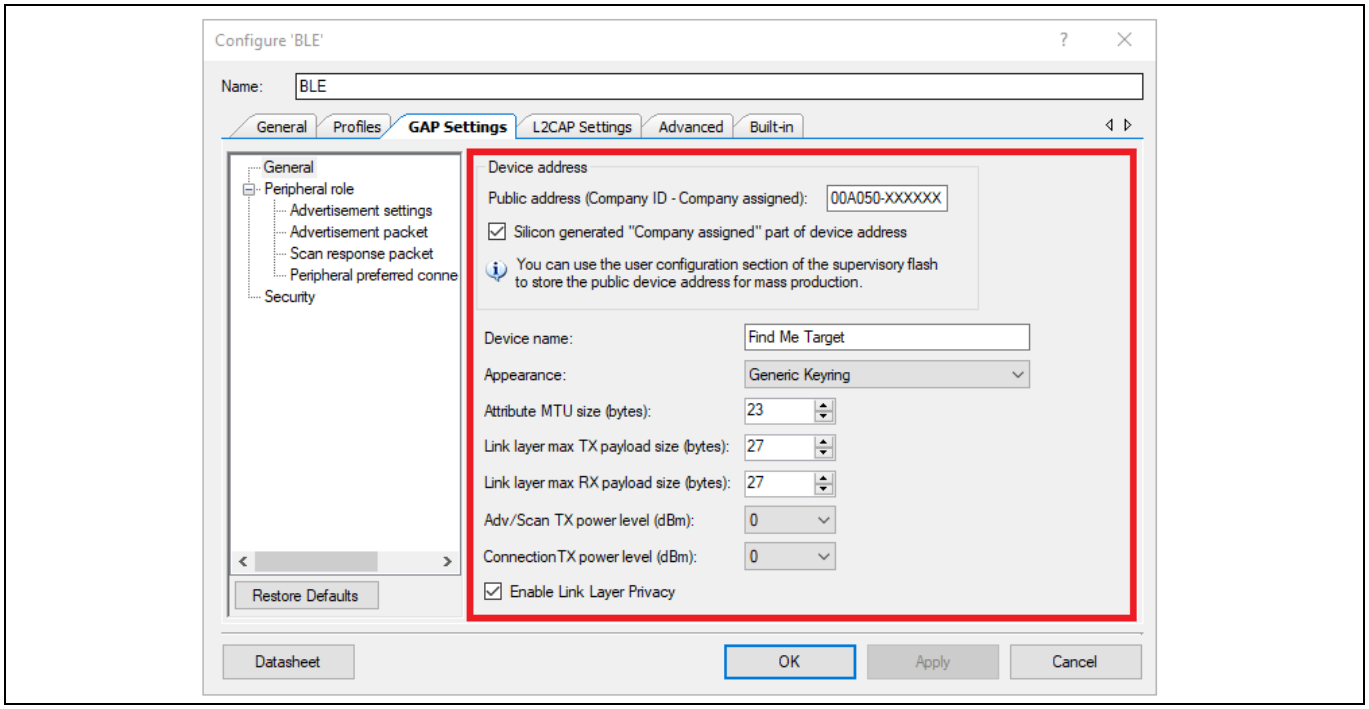


Figure 29 Bluetooth® LE 组件 GAP 通用设置

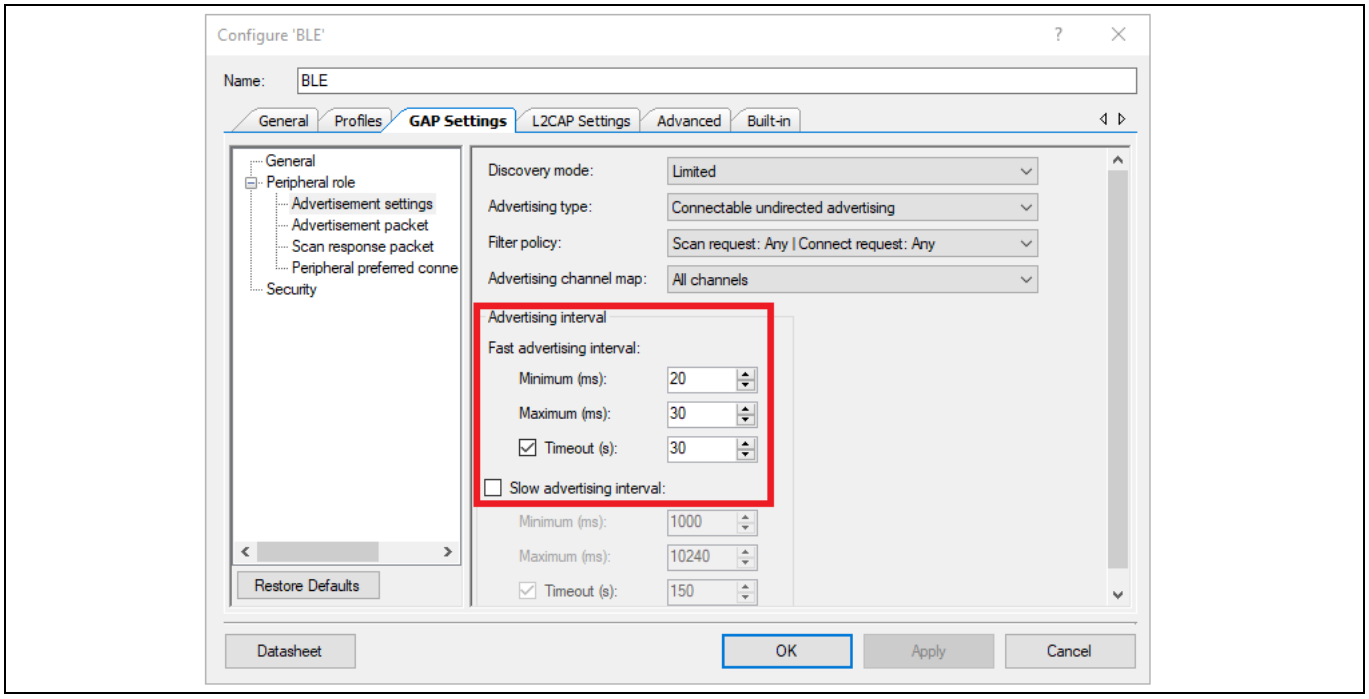


Figure 30 Bluetooth® LE 组件 GAP 广告设置

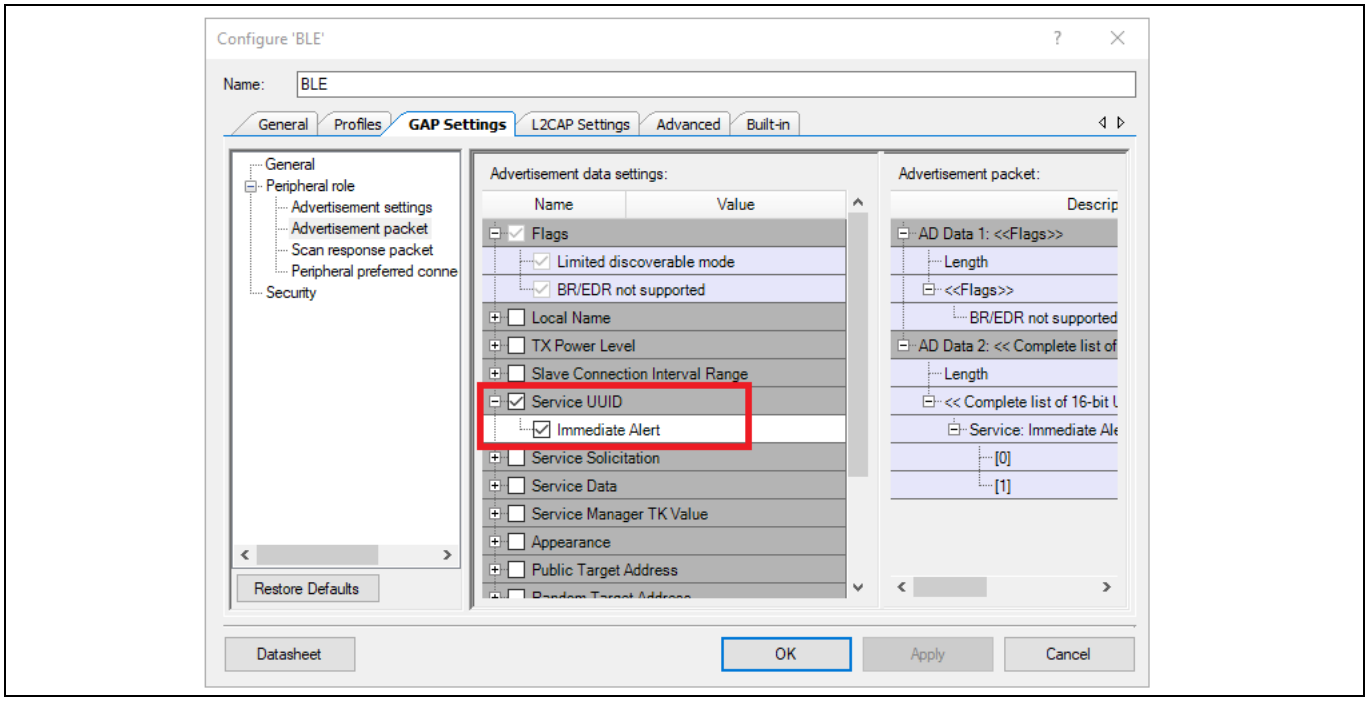


Figure 31 Bluetooth® LE 组件 GAP 广告数据包

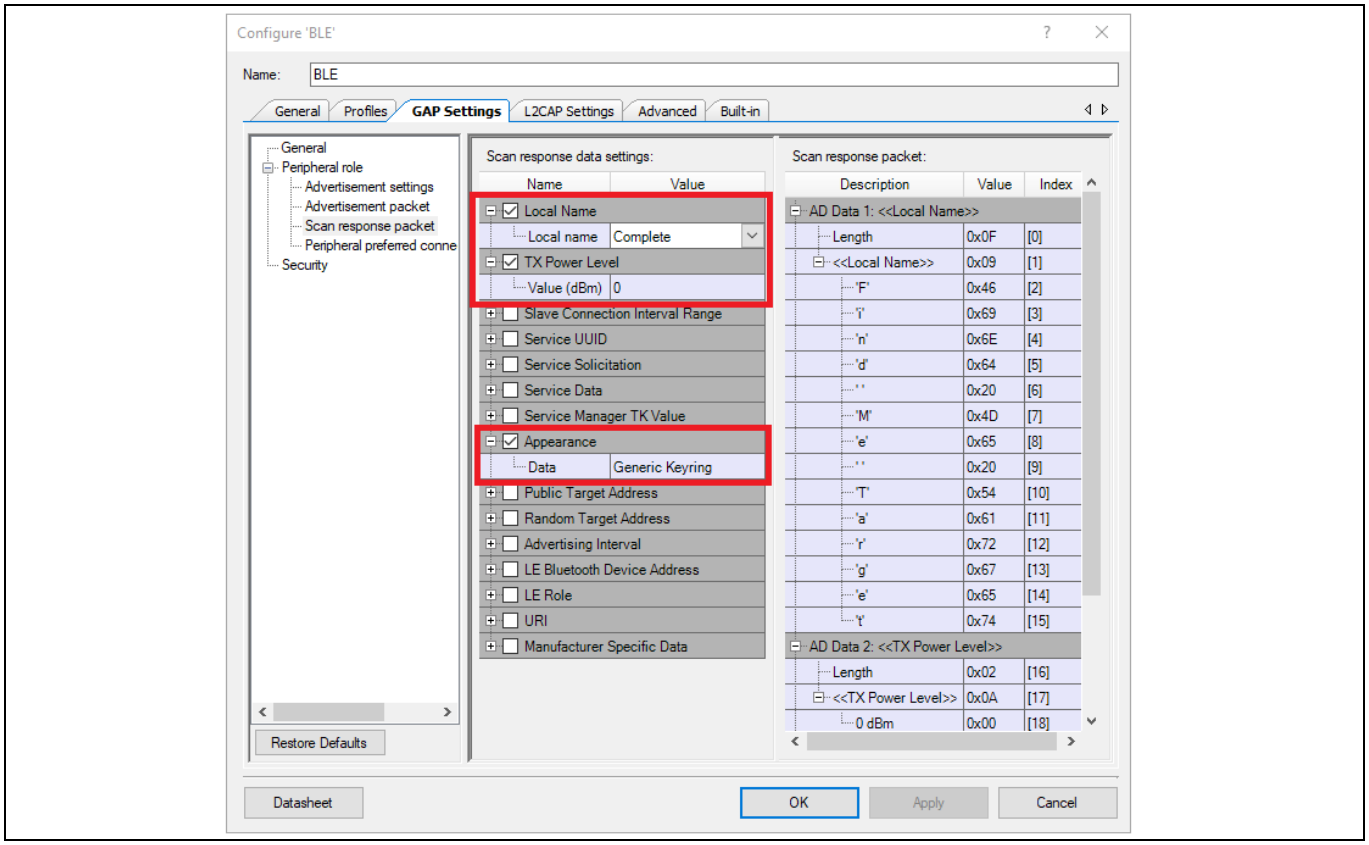


Figure 32 Bluetooth® LE 组件 GAP 扫描响应数据包

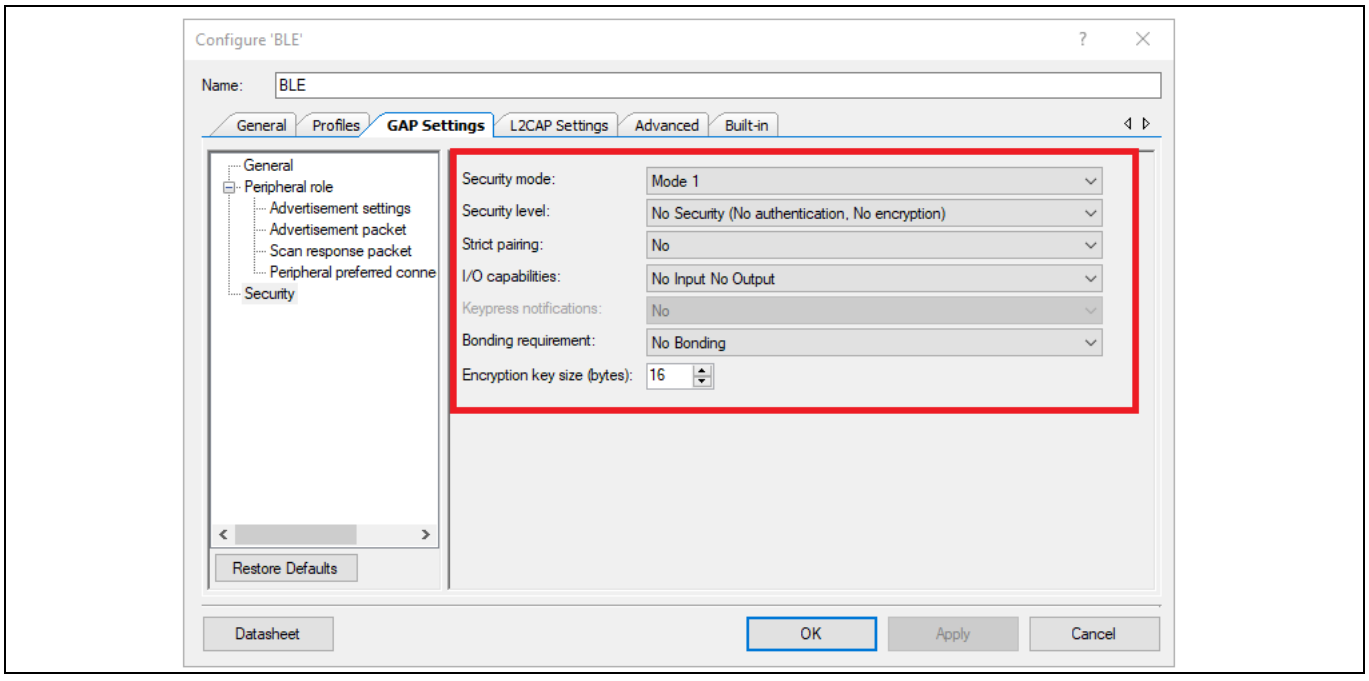


Figure 33 Bluetooth® LE 组件 GAP 安全设置

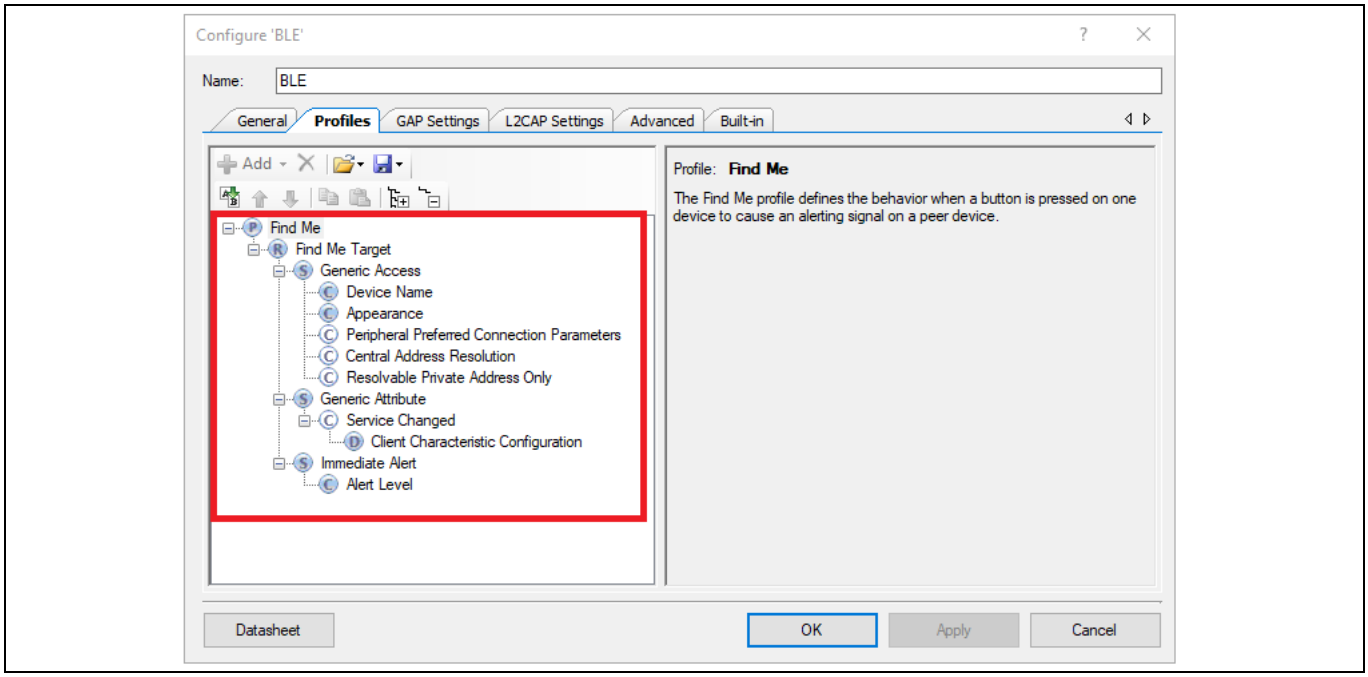


Figure 34　Bluetooth® LE 组件配置文件的配置

10. 拖放一个数字输出引脚组件，如 **Figure 35** 所示。

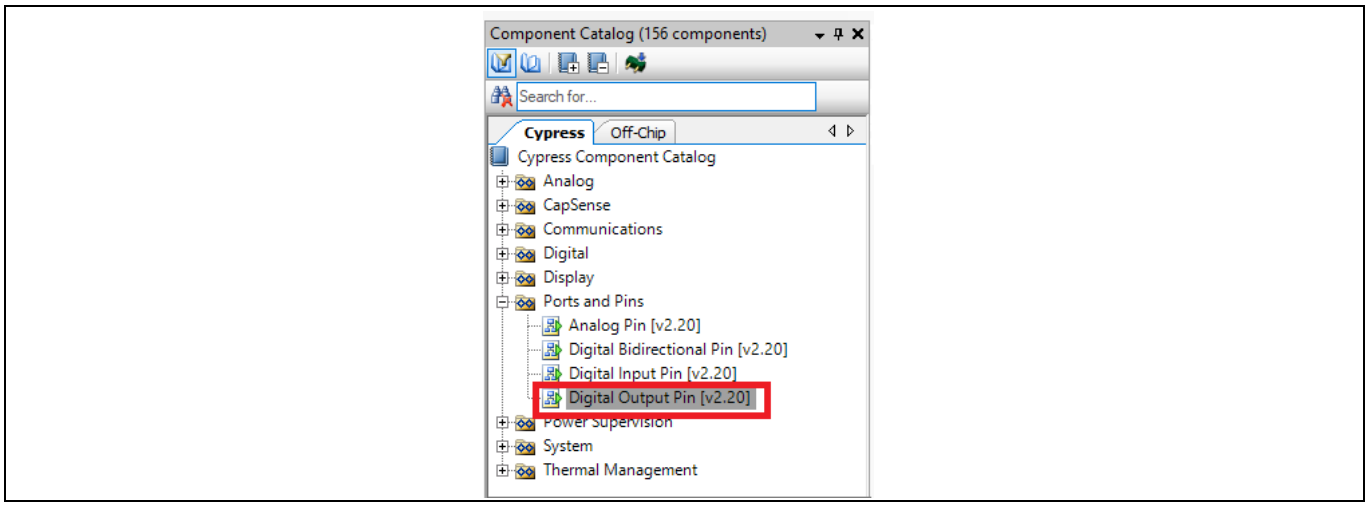


Figure 35　数字输出引脚组件的位置

11. 双击组件并进行以下更改，如 **Figure 36** 所示：

- 取消选择 **HW Connection**。
- 将名称改为 Alert_LED。
- 将 Initial drive state 设为“High(1)”。

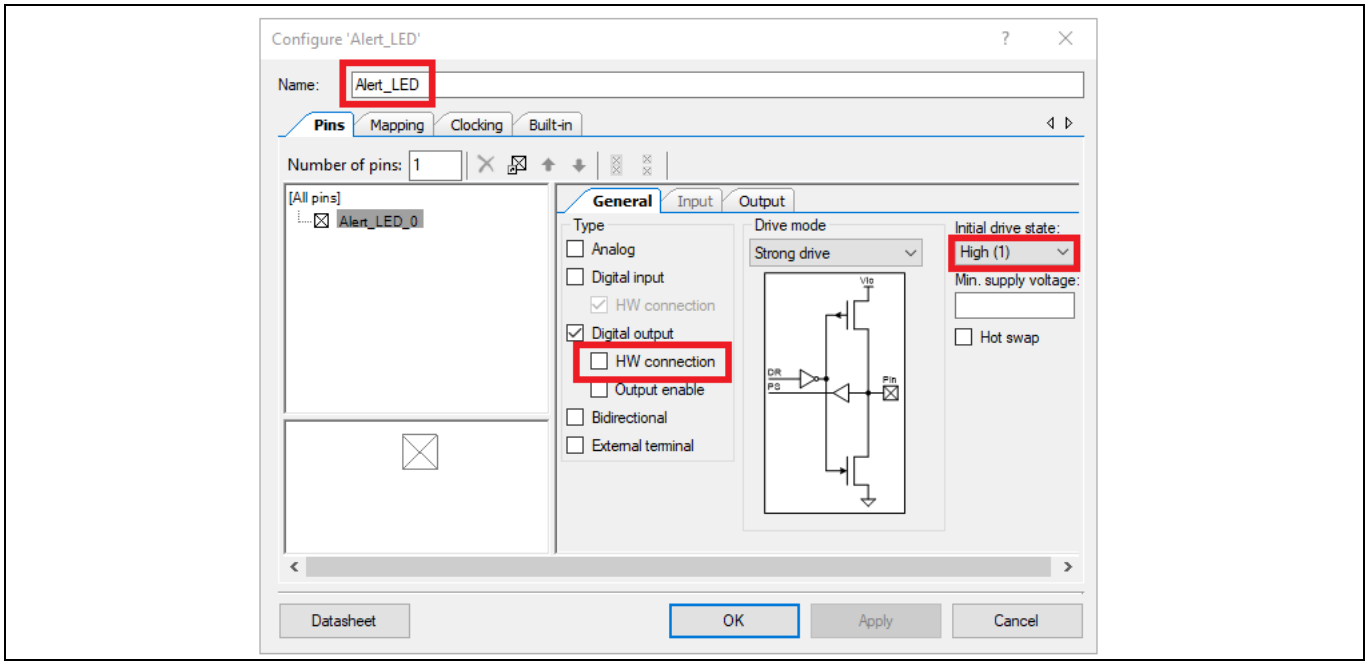


Figure 36　重新命名引脚组件

12. 添加两个数字输出引脚组件。双击组件并进行以下更改，如 **Figure 37** 和 **Figure 38** 所示：

- 将组件重新命名为 Advertising_LED 和 Disconnect_LED。
- 取消选择 HW Connection。
- 将 Initial drive state 设为“High(1)”。

这些引脚用于驱动 Bluetooth® LE 广告和断开连接状态指示灯 LED。Bluetooth® LE Pioneer 套件上的 LED 被设置为低电平有效，这样驱动 LED 的引脚状态为高电平时将关闭 LED，而其状态为低电平时会打开 LED。

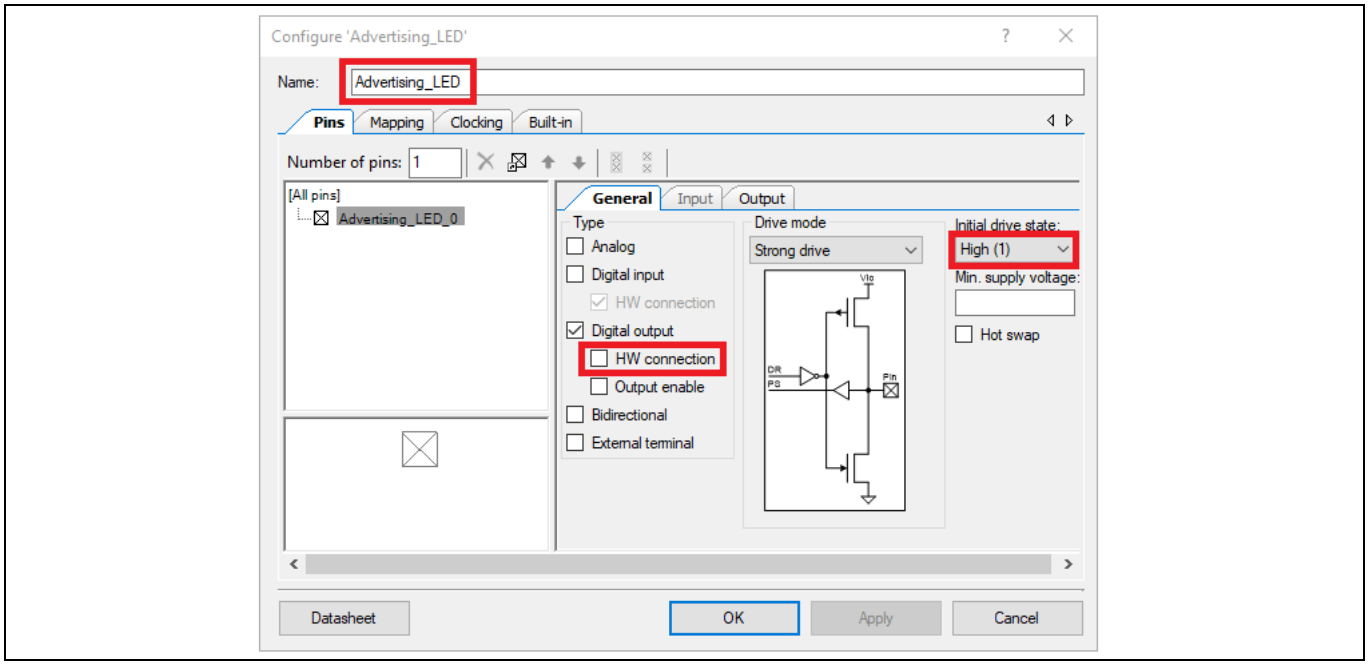


Figure 37　广告 LED 配置

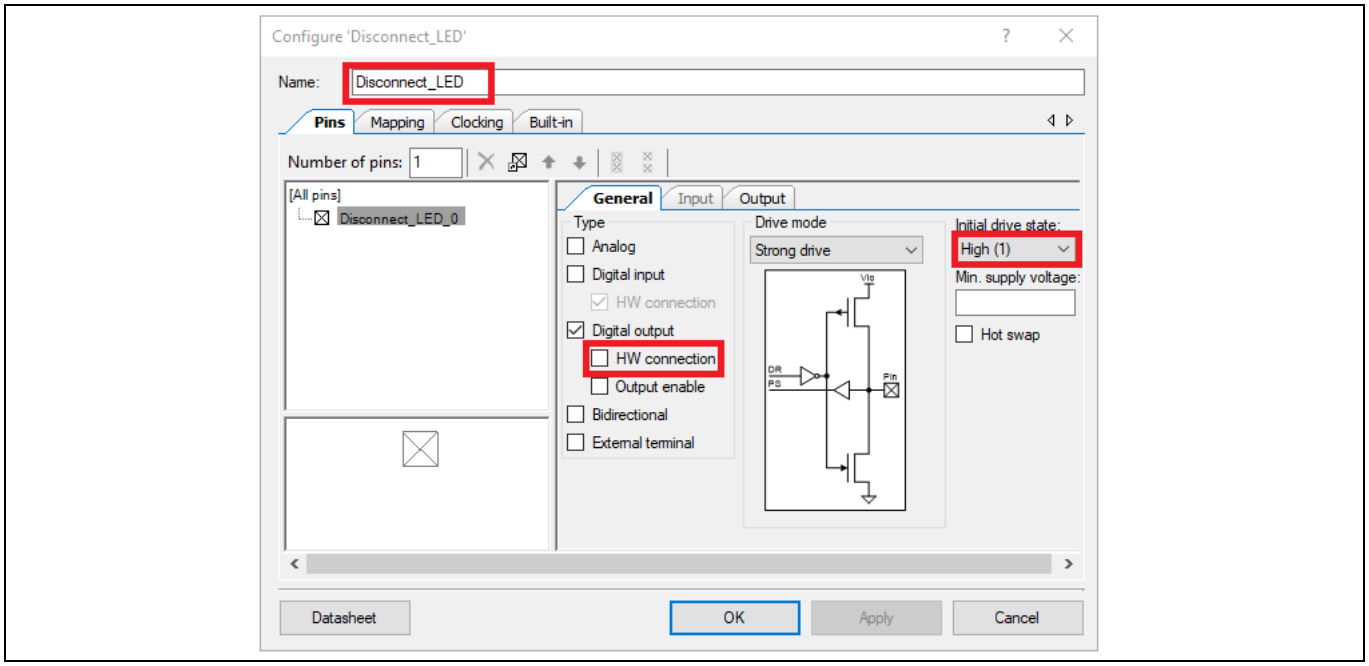


Figure 38　断开连接 LED 配置

13. 完成原理图配置后，您的设计将如 **Figure 39** 所示。

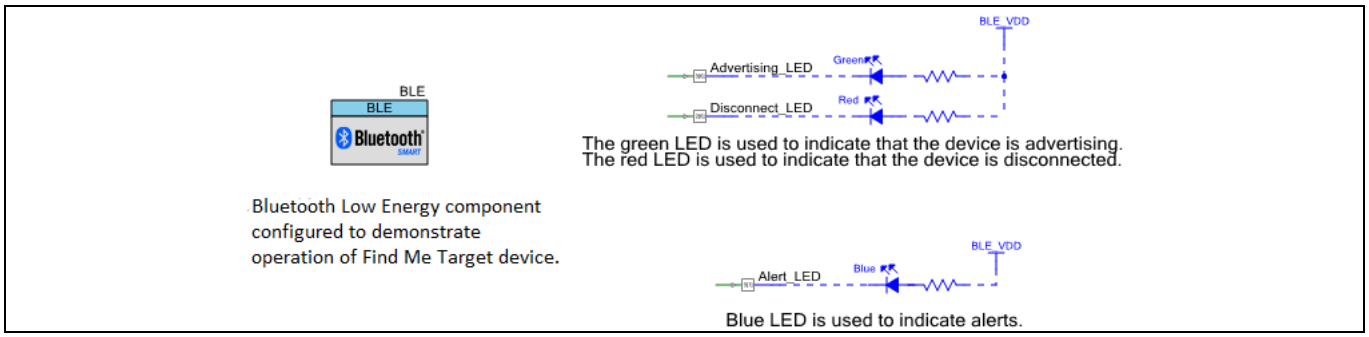


Figure 39　原理图配置

Note: ***Figure 39*** *中显示的蓝色虚线、LED 符号和电阻符号均表示片外 PSoC™ Creator 组件，仅用于说明信息，在您的设计中可有可无。如果想在设计中添加片外组件，可从 PSoC™ Creator 的片外组件目录中将所需组件拖放到您的设计原理图中。*

14. 在 **Workspace Explorer** 的 Design-Wide Resources (*BLE_FindMe.cydwr*) 文件下，双击 **Pins** 部分。您可以使用此部分为输出 (Advertising_LED、Disconnect_LED 和 Alert_LED) 选择器件引脚。**Figure 40** 显示了引脚配置内容，用于将 Advertising_LED、Disconnect_LED 和 Alert_LED 引脚分别连接到 Bluetooth® LE Pioneer 套件中绿色、红色和蓝色 LED 上。

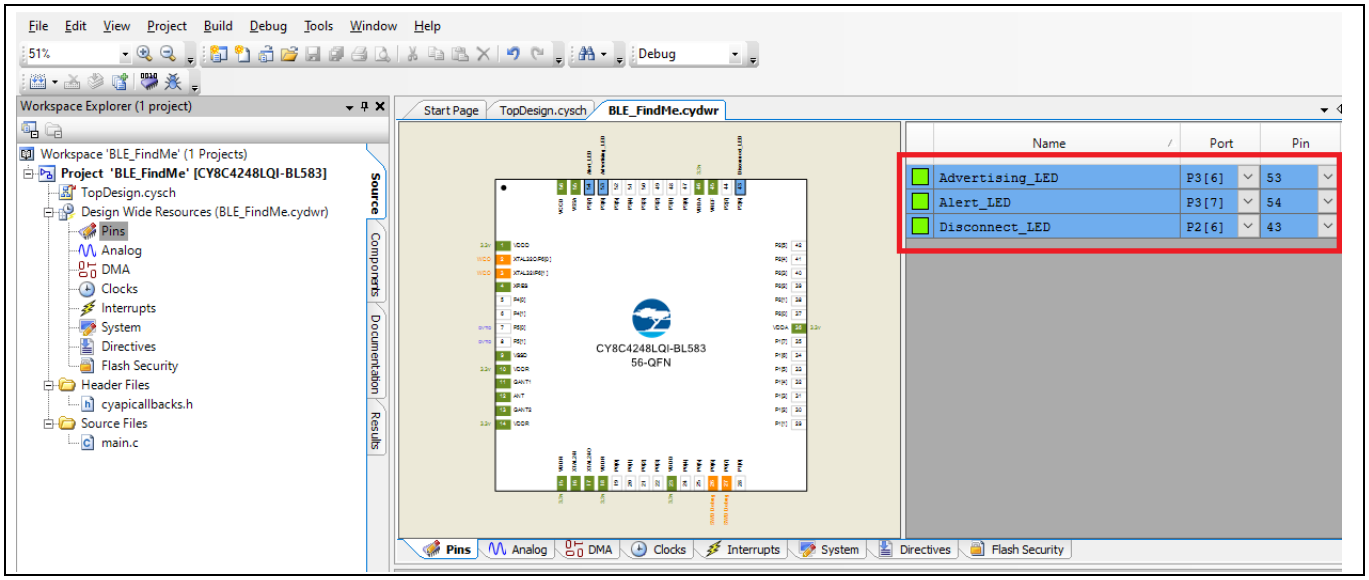


Figure 40　引脚选择

如果您正在使用自己的电路板或某个不带 LED 的开发套件，那么请选择相关引脚。您可以将外部 LED 连接到选中的引脚上，如 **Figure 21** 所示。

15. 同样，在 Design Wide Resources (BLE_FindMe.cydwr) 文件下的 **Clocks** 部分，单击 **Edit Clock** 按钮。将打开一个配置时钟的窗口。将打开一个配置时钟的窗口。转到 **High Frequency Clocks** (高频时钟) 选项卡并修改 IMO 以将内部主振荡器 (IMO) 配置为 12 MHz，如 **Figure 41** 所示。单击 **OK**。

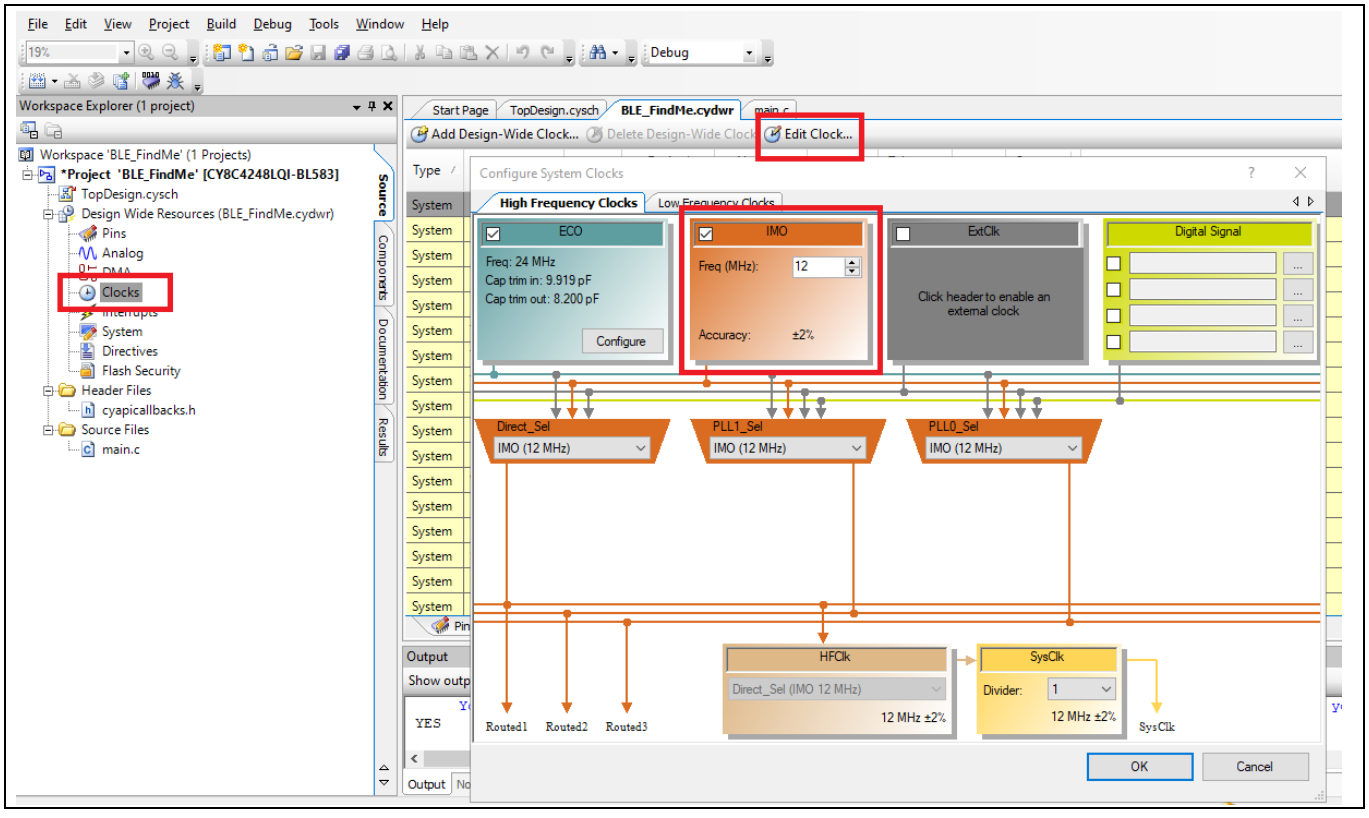


Figure 41　时钟配置

16. 从 **Build** 菜单中选择 **Generate Application**。注意：在 **Workspace Explorer** 窗口中，PSoC™ Creator 将自动为 Bluetooth® LE 和数字输出引脚组件生成源代码文件，如 **Figure 42** 所示。转到下一节前，请清除所生成 main.c 文件中的内容。

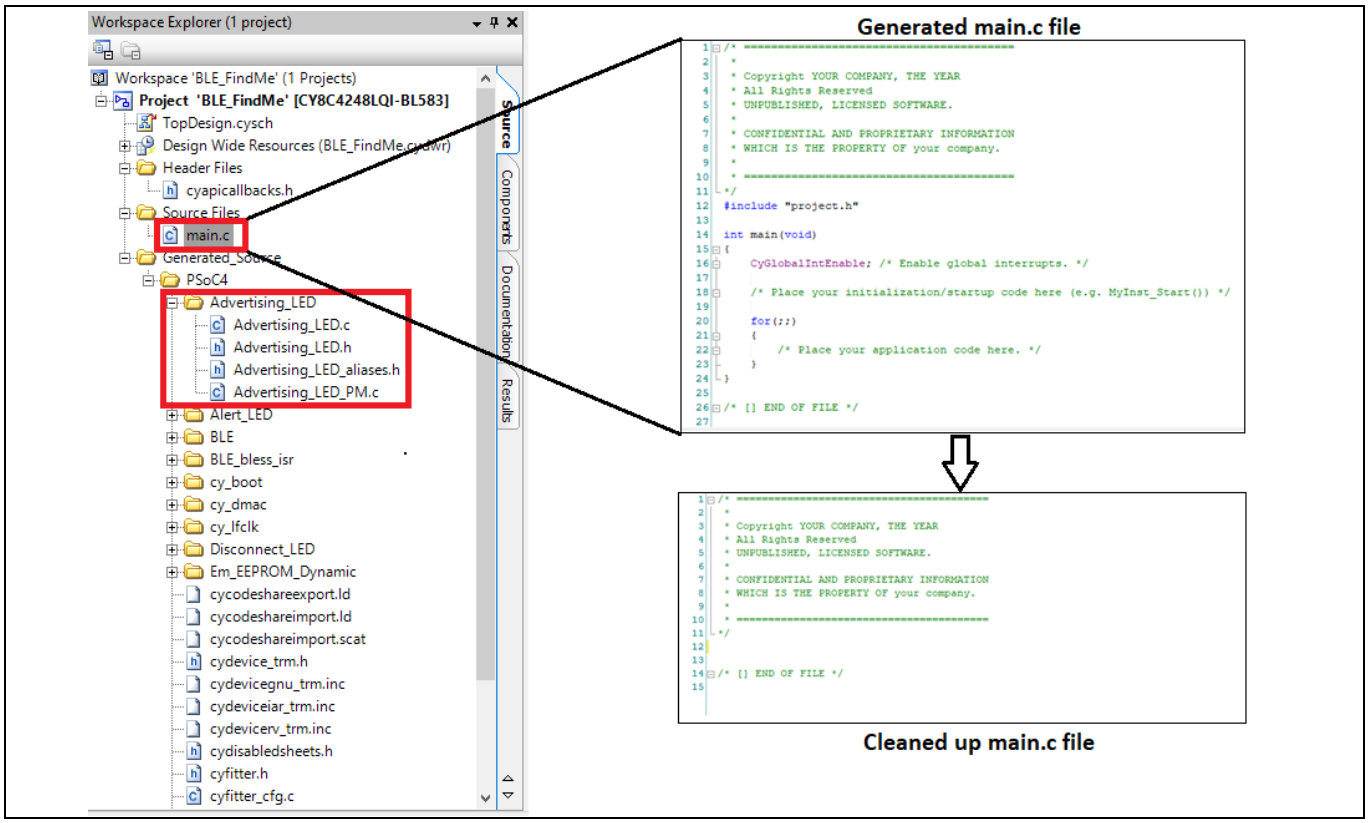


Figure 42　所生成的源文件

7.4　第二步：编写固件

使用 PSoC™ Creator 设计 Bluetooth® LE 标准配置文件应用时需要以下四个主要的固件模块：

- 系统初始化
- Bluetooth® LE 堆栈事件处理程序
- Bluetooth® LE 特定服务事件处理程序
- 主循环和低功耗操作

下面各节将讨论这些模块，讨论内容与您在**第一步：构建设计**中所配置的设计相关。本节所提供的代码段存储在 **Figure 42** 中的 main.c 文件内。

7.4.1　系统初始化

PSoC™ 4 with AIROC™ Bluetooth® LE 器件被复位时，固件首先会初始化系统，包括使能全局中断和使能设计中所用的其他组件。系统被初始化后，固件将初始化 Bluetooth® LE 组件，该操作会内部初始化整个 Bluetooth® LE 子系统。

在进行 Bluetooth® LE 组件初始化过程中，您需要传送事件处理函数，Bluetooth® LE 堆栈将调用该函数来通知待处理事件。**Code Listing 2** 中所示的 Bluetooth® LE 堆栈事件处理程序被注册为 Bluetooth® LE 初始化过程的一部分。如果成功初始化了 Bluetooth® LE 组件，固件将为即时警报服务 (IAS) 的特定事件注册其他事

件处理程序，并将控制权交给主循环。**Figure 43** 和 **Code Listing 1** 显示了系统初始化的流程图和固件源代码。

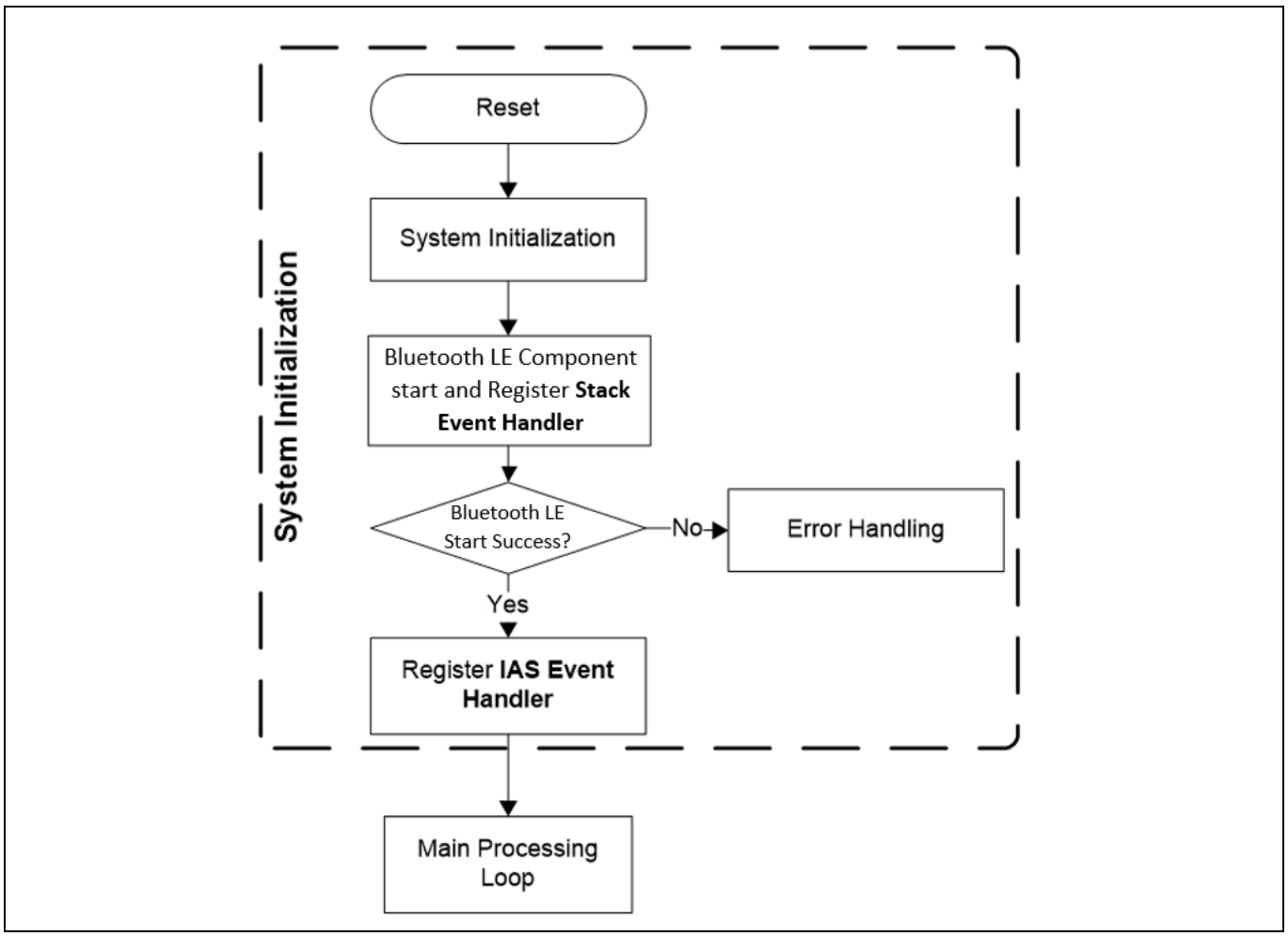


Figure 43　系统初始化流程图

Code Listing 1　系统初始化固件

```
#include <project.h>

#define LED_ON                  (0u)
#define LED_OFF                 (1u)

#define NO_ALERT                (0u)
#define MILD_ALERT              (1u)
#define HIGH_ALERT              (2u)

#define LED_TOGGLE_TIMEOUT      (100u)
```

Code Listing 1　　系统初始化固件

```
void StackEventHandler(uint32 event, void *eventParam);
void IasEventHandler(uint32 event, void *eventParam);

uint8 alertLevel;

int main()
{
  CYBLE_API_RESULT_T apiResult;

  CyGlobalIntEnable;

  apiResult = CyBle_Start(StackEventHandler);

  if(apiResult != CYBLE_ERROR_OK)
  {
    /* BLE stack initialization failed, check your configuration */
    CYASSERT(0);
  }

  CyBle_IasRegisterAttrCallback(IasEventHandler);

  /* Place the main application loop here */
}
```

7.4.2　Bluetooth® LE 堆栈事件处理程序

Bluetooth® LE 组件中的 Bluetooth® LE 堆栈会生成一系列事件，用于通过您注册的 Bluetooth® LE 堆栈事件处理程序将 Bluetooth® LE 接口状态和数据提供给应用固件。事件处理程序必须处理堆栈所生成的几种基本事件 (如器件连接、堆栈初始化成功完成) 并准确配置堆栈，从而能够成功建立和维持 Bluetooth® LE 连接。对于我们所创建的“Find Me Target”应用，Bluetooth® LE 堆栈事件处理程序必须处理 **Table 1** 中所描述的所有事件。**Figure 44** 和 **Code Listing 2** 显示了处理 Bluetooth® LE 堆栈事件的流程图和固件。

Table 1 Bluetooth® LE 堆栈事件

Bluetooth® LE 堆栈事件名称	事件说明	事件处理程序操作
CYBLE_EVT_STACK_ON	成功完成 Bluetooth® LE 堆栈初始化	开始广告并将广告状态反映在 LED 上
CYBLE_EVT_GAP_DEVICE_DISCONNECTED	断开了与对等器件的 Bluetooth® LE 连接	重启广告并将广告状态反映在 LED 上
CYBLE_EVT_GAP_DEVICE_CONNECTED	已经建立好了与对等器件的 Bluetooth® LE 连接	更新了 LED 上的 Bluetooth® LE 连接状态
CYBLE_EVT_GAPP_ADVERTISEMENT_START_STOP	Bluetooth® LE 堆栈广告启用/停止事件	如果该广告已过期，会将器件设置为停止模式

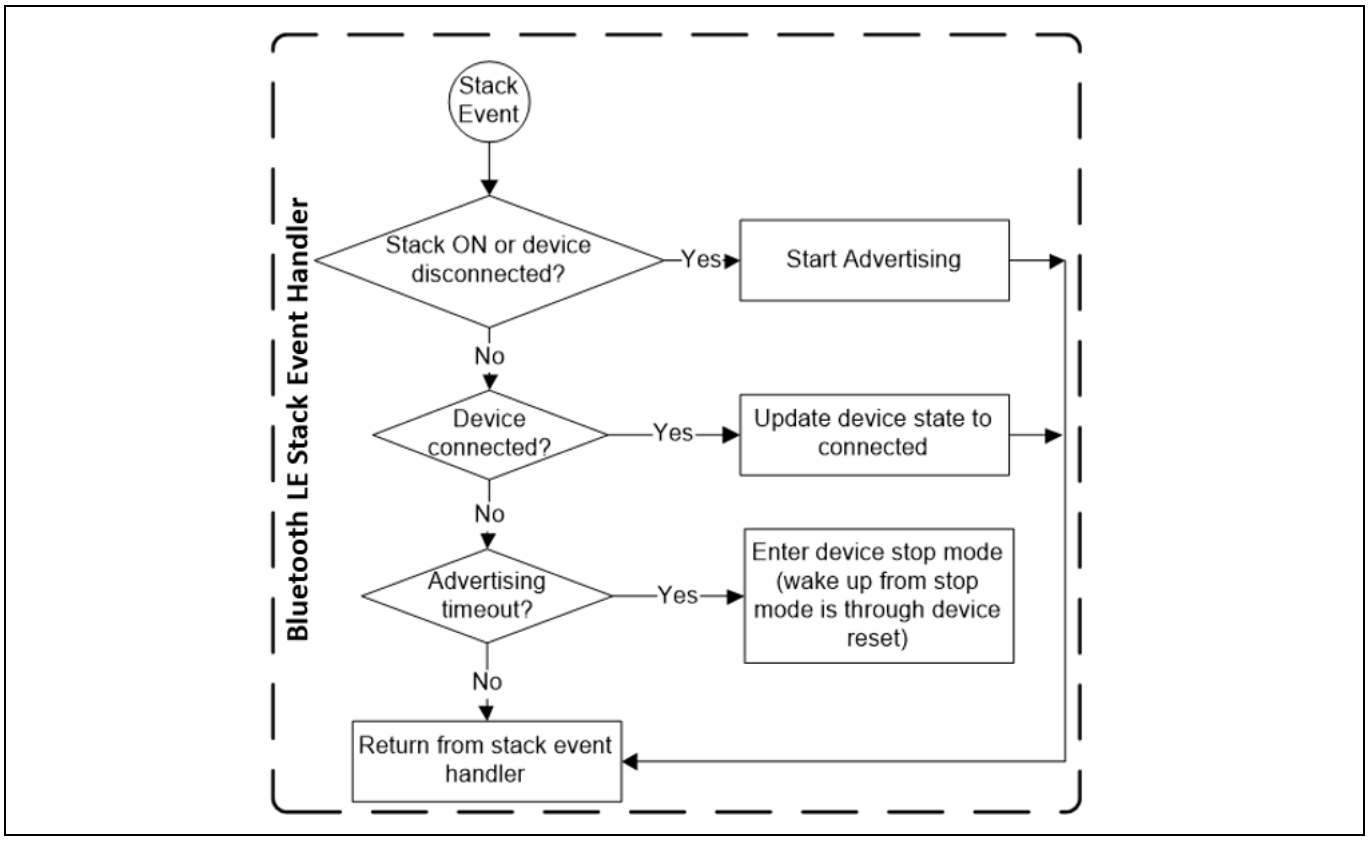


Figure 44 Bluetooth® LE 堆栈事件处理程序流程图

Code Listing 2 Bluetooth® LE 堆栈事件处理程序固件

```
void StackEventHandler(uint32 event, void *eventParam)
{
    switch(event)
    {
        /* Mandatory events to be handled by Find Me Target design */
        case CYBLE_EVT_STACK_ON:
        case CYBLE_EVT_GAP_DEVICE_DISCONNECTED:
```

Code Listing 2　Bluetooth® LE 堆栈事件处理程序固件

```
            /* Start BLE advertisement for 30 seconds and update link
             * status on LEDs */
            CyBle_GappStartAdvertisement(CYBLE_ADVERTISING_FAST);
            Advertising_LED_Write(LED_ON);
            alertLevel = NO_ALERT;
        break;

        case CYBLE_EVT_GAP_DEVICE_CONNECTED:
            /* BLE link is established */
            Advertising_LED_Write(LED_OFF);
            Disconnect_LED_Write(LED_OFF);
        break;

        case CYBLE_EVT_GAPP_ADVERTISEMENT_START_STOP:
            if(CyBle_GetState() == CYBLE_STATE_DISCONNECTED)
            {
                /* Advertisement event timed out, go to low power
                 * mode (Stop mode) and wait for device reset
                 * event to wake up the device again */
                Advertising_LED_Write(LED_OFF);
                Disconnect_LED_Write(LED_ON);
                CySysPmSetWakeupPolarity(CY_PM_STOP_WAKEUP_ACTIVE_HIGH);
                CySysPmStop();

                /* Code execution will not reach here */
            }
        break;

        default:
        break;
    }
}
```

7.4.3 Bluetooth® LE 特定服务事件处理程序

Bluetooth® LE 组件还生成与您设计所支持的每个服务相对应的事件。对于我们正在创建的“Find Me Target”应用，Bluetooth® LE 组件将生成 IAS 事件，以通知应用程序警报级别特性是否要被更新为某个新值。**Figure 45** 和 **Code Listing 3** 分别显示了处理 Bluetooth® LE IAS 事件的流程图和固件。

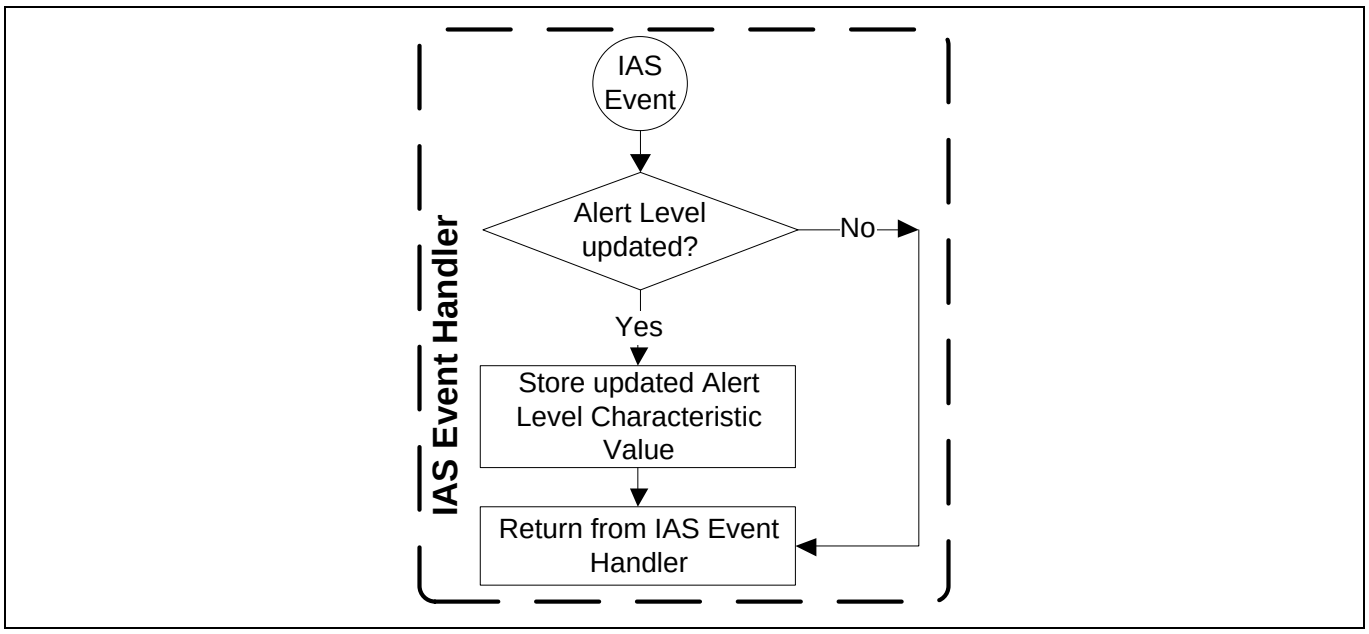


Figure 45 Bluetooth® LE IAS 事件处理程序流程图

Code Listing 3 Bluetooth® LE IAS 事件处理程序固件

```
void IasEventHandler(uint32 event, void *eventParam)
{
    /* Alert Level Characteristic write event */
    if(event == CYBLE_EVT_IASS_WRITE_CHAR_CMD)
    {
        /* Read the updated Alert Level value from the GATT database */
        CyBle_IassGetCharacteristicValue(CYBLE_IAS_ALERT_LEVEL,
            sizeof(alertLevel), &alertLevel);
    }
}
```

7.4.4 主循环和低功耗操作

您正在设计的主循环固件必须定期服务 Bluetooth® LE 堆栈处理事件，根据每个 IAS 警报级别的特性值更新蓝色报警 LED 的状态，并将 **Bluetooth® LE 子系统 (BLESS)** 模块和 PSoC™ 4 with AIROC™ Bluetooth® LE 器件配置为低功耗模式。**Figure 46** 和 **Code Listing 4** 显示了主循环的流程图和固件。在注册 IasEventHandler 函数回调后，您应该将 **Code Listing 4** 中的整个代码添加到 **Code Listing 1** 中所显示的主函数内。

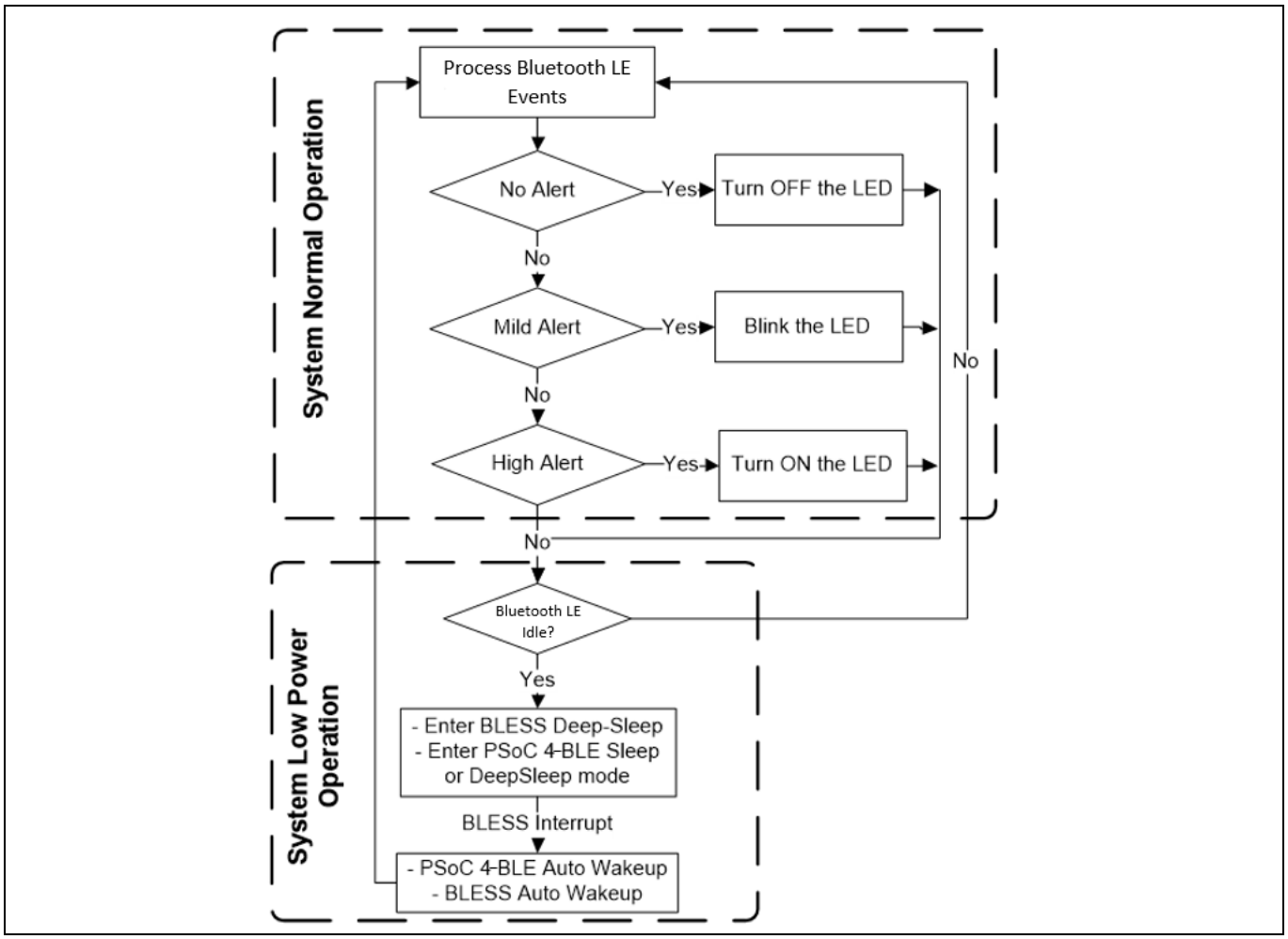


Figure 46 固件主循环流程图

Code Listing 4 主循环固件

```
/* Place the main application loop here */

for(;;)
{
    static uint8 toggleTimeout = 0;
    CYBLE_BLESS_STATE_T blessState;
    uint8 intrStatus;
```

Code Listing 4　主循环固件

```
    /* Single API call to service all the BLE stack events. Must be
     * called at least once in a BLE connection interval */
                                        CyBle_ProcessEvents();
    /* Update Alert Level value on the blue LED */
    switch(alertLevel)
    {
        case NO_ALERT:
        Alert_LED_Write(LED_OFF);
        break;

        case MILD_ALERT:
        toggleTimeout++;
        if(toggleTimeout == LED_TOGGLE_TIMEOUT)
        {
            /* Toggle alert LED after timeout */
            Alert_LED_Write(Alert_LED_Read() ^ 0x01);
            toggleTimeout = 0;
        }
        break;

        case HIGH_ALERT:
        Alert_LED_Write(LED_ON);
        break;
    }

    /* Configure BLESS in Deep-Sleep mode */
    CyBle_EnterLPM(CYBLE_BLESS_DEEPSLEEP);

    /* Prevent interrupts while entering system low power modes */
    intrStatus = CyEnterCriticalSection();

    /* Get the current state of BLESS block */
    blessState = CyBle_GetBleSsState();

    /* If BLESS is in Deep-Sleep mode or the XTAL oscillator is turning on,
```

Code Listing 4　主循环固件

```
     * then PSoC™ 4 BLE can enter Deep-Sleep mode (1.3uA current consumption) */
    if(blessState == CYBLE_BLESS_STATE_ECO_ON ||
        blessState == CYBLE_BLESS_STATE_DEEPSLEEP)
    {
        CySysPmDeepSleep();
    }
    else if(blessState != CYBLE_BLESS_STATE_EVENT_CLOSE)
    {
        /* If BLESS is active, then configure PSoC™ 4 BLE system in
         * Sleep mode (~1.6mA current consumption) */
        CySysPmSleep();
    }
    else
    {
        /* Keep trying to enter either Sleep or Deep-Sleep mode */
    }
    CyExitCriticalSection(intrStatus);

    /* BLE link layer timing interrupt will wake up the system from Sleep
     * and Deep-Sleep modes */
}
```

7.5　第三步：编程器件

本部分介绍了如何对 PSoC™ 4 with AIROC™ Bluetooth® LE 器件进行编程。如果您正在使用带有内置编程器的开发套件 (如 Bluetooth® LE Pioneer 套件)，请使用 USB 线缆将该开发板连接到您的电脑上。对于其他套件，请参考相应的套件指南。如果您正在使用自己的硬件进行开发，那么需要使用一个硬件调试器，例如，Infineon **CY8CKIT-002 MiniProg3**。

Note:　*本设计的源项目位于 PSoC™ Creator 3.3 SP1 或更高版本的下面路径：File > Code Example，将 Filter by 选为 Find Me> BLE_FindMe。*

1. 在 PSoC™ Creator 中，选择 **Debug** > **Select Debug Target**，如 **Figure 47** 所示。

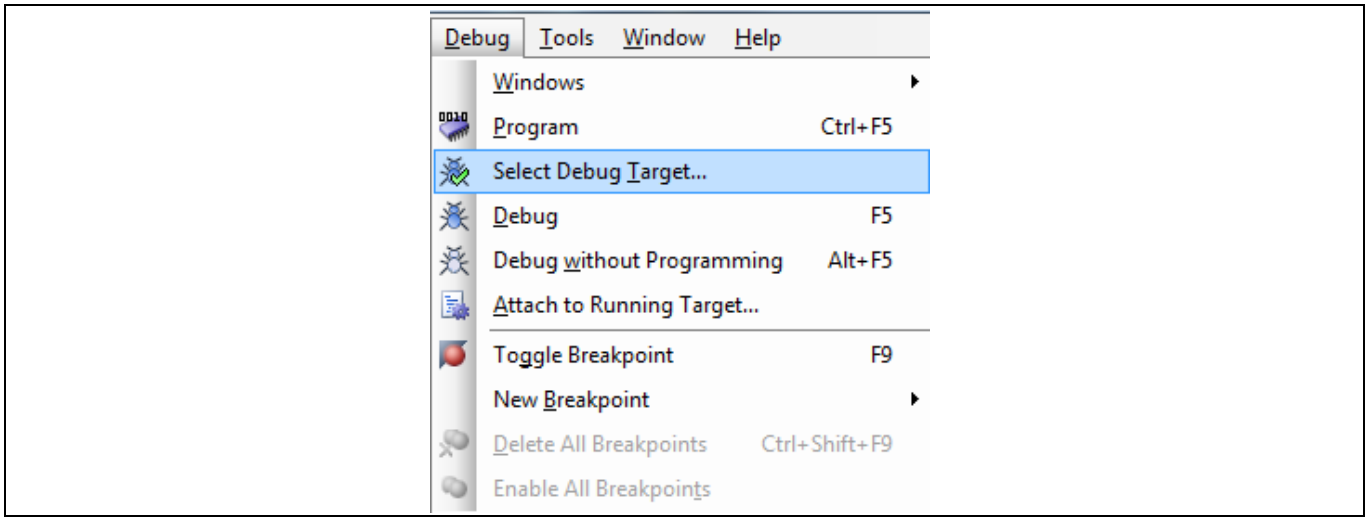


Figure 47　选择调试对象

2. 在 **Select Debug Target** 对话框中，单击 **Port Acquire**，然后选择 **Connect**，如 **Figure 48** 所示。单击 **OK** 以关闭对话框。

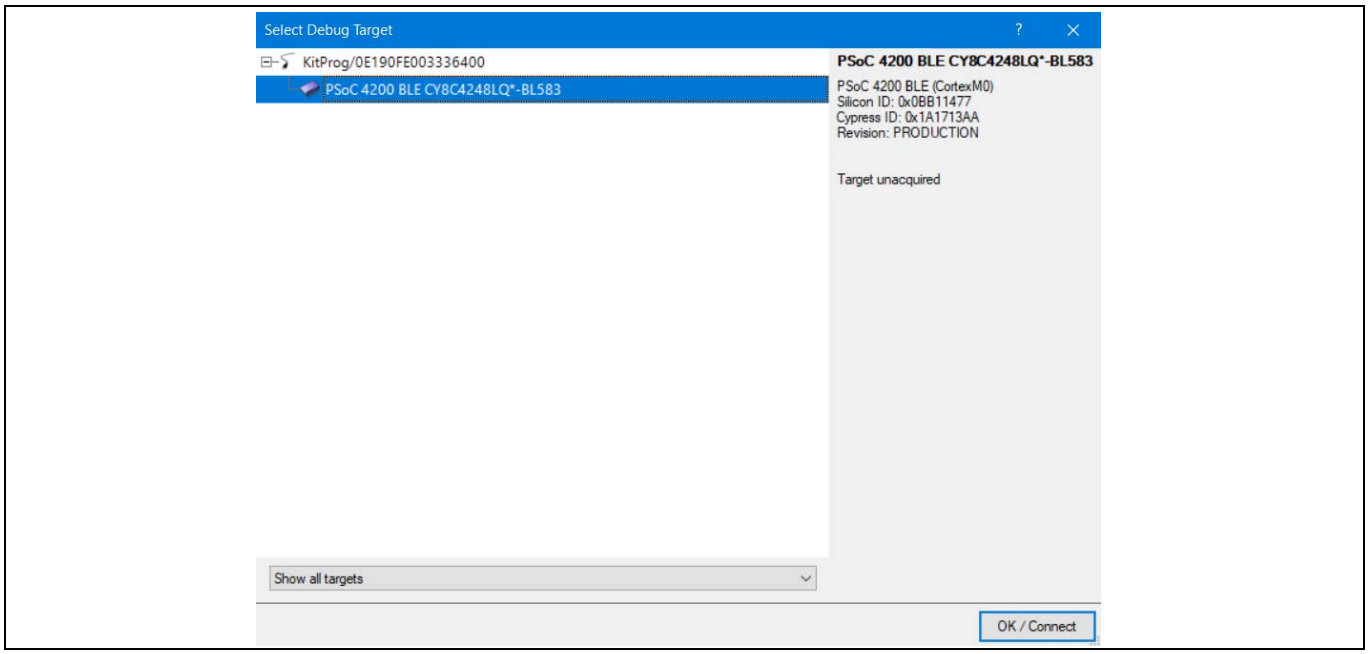


Figure 48　连接到器件

如果您正在使用自己的硬件，请确保 **Select Debug Target** 窗口中用于编程硬件的 **Port Setting** 配置符合您的设置。

3. 在 PSoC™ Creator 中，依次选择 **Debug** > **Program**，从而将项目编程到器件内，如 **Figure 49** 所示。

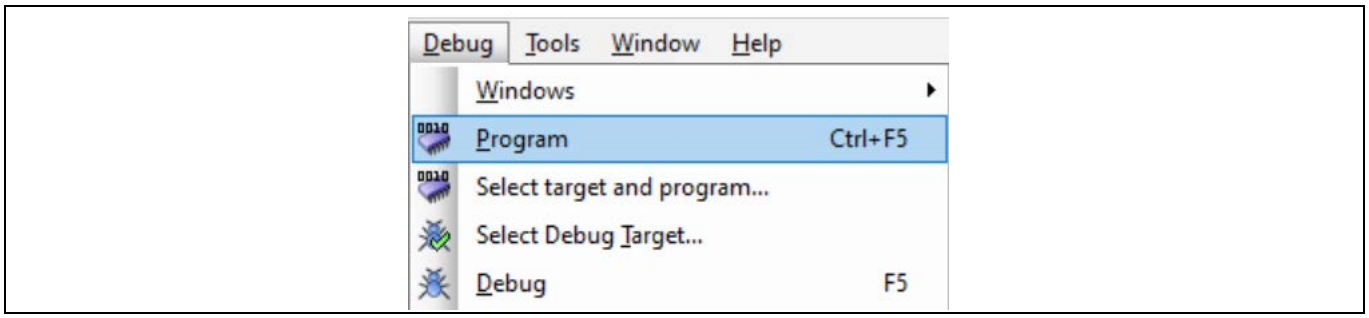


Figure 49 编程器件

4. 可在 PSoC™ Creator 状态栏上 (窗口的左下角) 查看编程状态，如 **Figure 50** 所示。

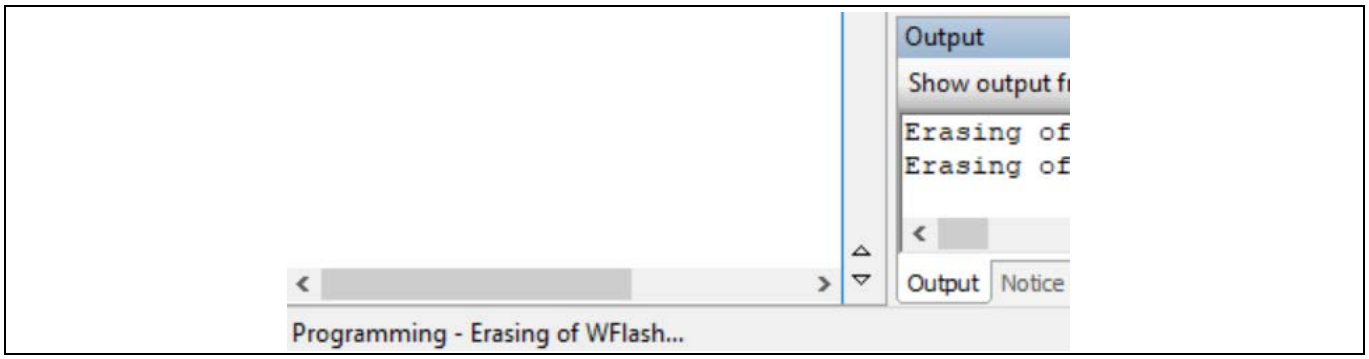


Figure 50 编程状态

7.6 第四步：测试您的设计

本部分介绍了如何使用各种 CySmart 手机应用和电脑工具来测试您的 Bluetooth® LE 设计。**Figure 19** 显示了使用 Bluetooth® LE Pioneer 套件测试设计的设置情况。

有关这些工具的详细信息，请参考以下各附录：

- **CySmart 主机仿真工具**
- **手机版的 CySmart 应用**

7.6.1 使用 CySmart 手机应用

1. 打开 iOS 或 Android 设备的蓝牙。
2. 启动 CySmart 应用。
3. 按下 Bluetooth® LE Pioneer 套件上的复位开关，以启动您设计上的 Bluetooth® LE 广告。
4. 下拉 CySmart 应用的主屏幕，从而开始扫描 Bluetooth® LE 外设，这时您的器件会出现在 CySmart 应用的主屏幕上。选择您的器件来建立 Bluetooth® LE 连接。
5. 从旋转视图中选择“Find Me”配置文件。
6. 在 Find Me **配置文件**屏幕上选择一个警报级别，然后根据所选级别观察设备上 LED 状态的变化。

Figure 51 和 **Figure 52** 分别显示了 CySmart 手机应用版的逐步配置屏幕截图。

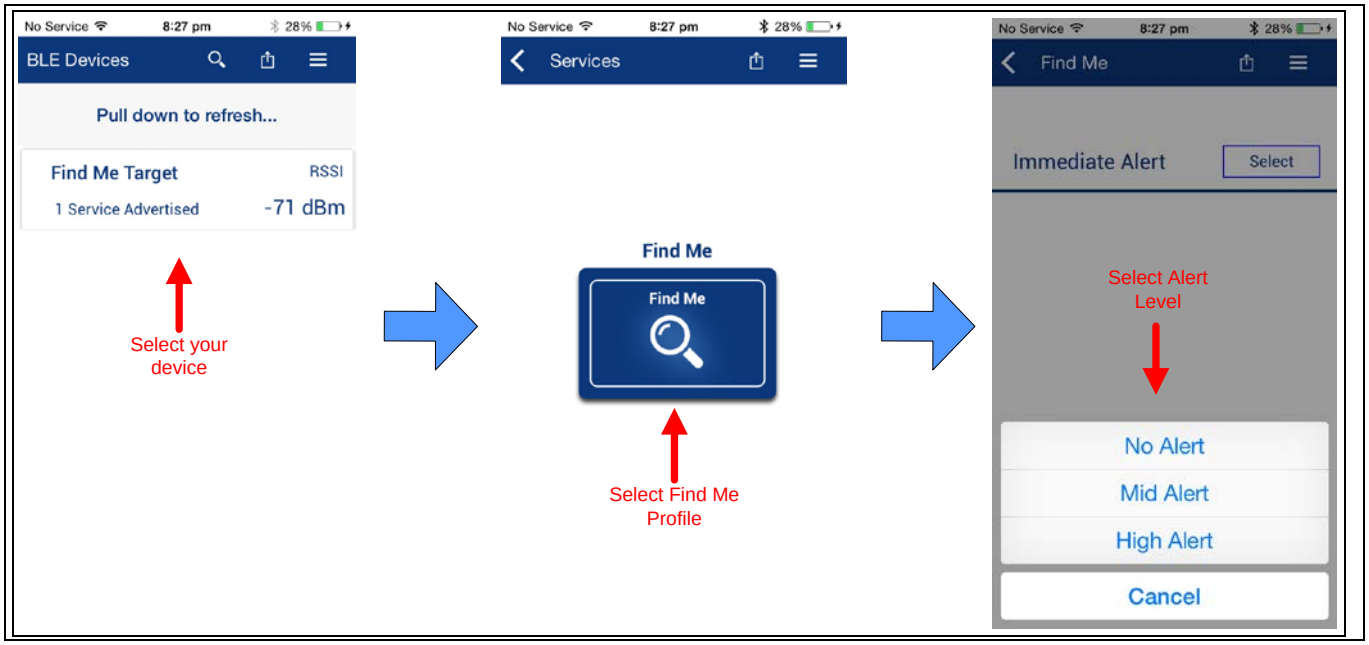


Figure 51　使用 CySmart iOS 应用进行测试

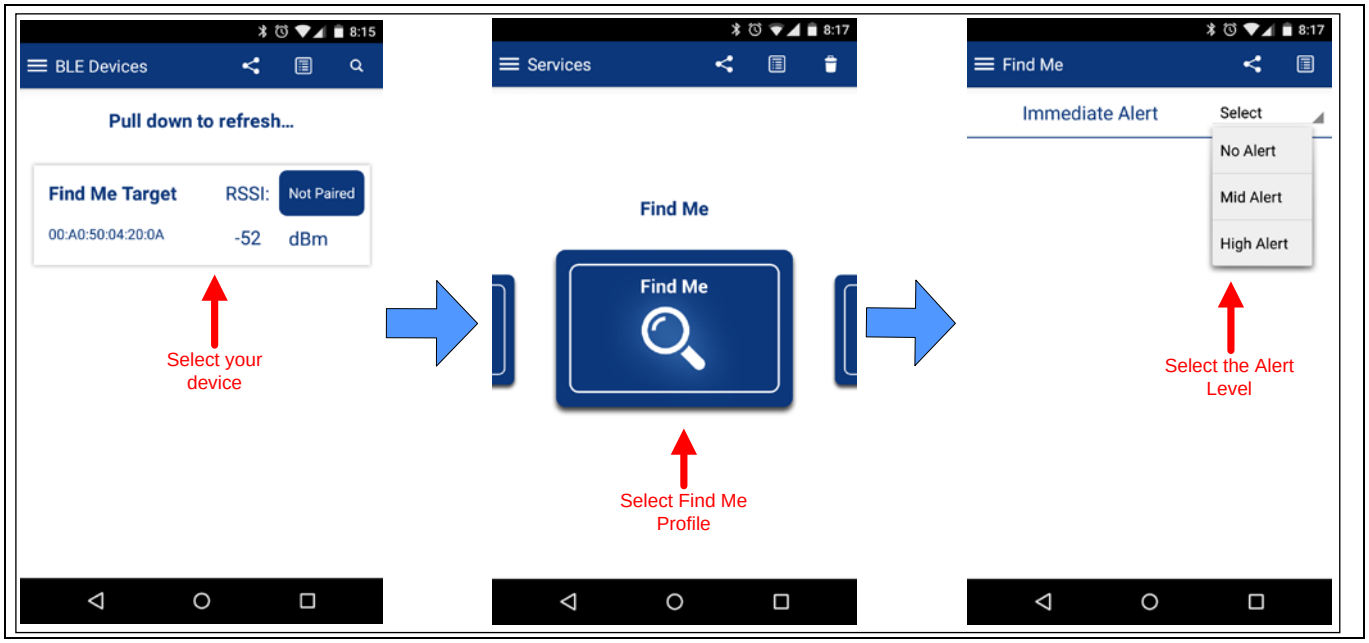


Figure 52　使用 CySmart Android 应用进行测试

7.6.2　使用 CySmart 电脑主机仿真工具

除了使用 CySmart 手机应用外，您还可以使用 CySmart 电脑主机仿真工具为您的设计建立 Bluetooth® LE 连接，并对 Bluetooth® LE 特性进行读或写操作。

1. 将 Bluetooth® LE 收发器连接到 Windows 电脑上。等待驱动程序安装完成。
2. 启动 CySmart 主机仿真工具，该工具会自动检测 Bluetooth® LE 收发器。如果在 **Select BLE Dongle Target** 弹出式窗口中尚未显示 Bluetooth® LE 收发器，请点击 **Refresh**。然后点击 **Connect**，如 **Figure 53** 所示。

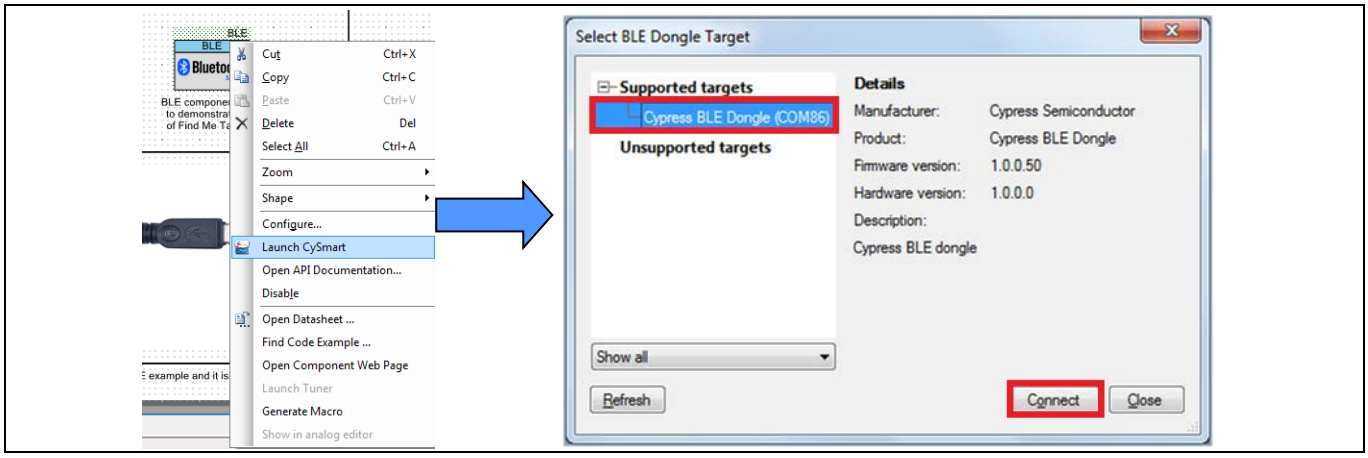


Figure 53 CySmart Bluetooth® LE 收发器选择

3. 选择 **Configure Master Settings**，并将各值恢复为默认设置，如 **Figure 54** 所示。

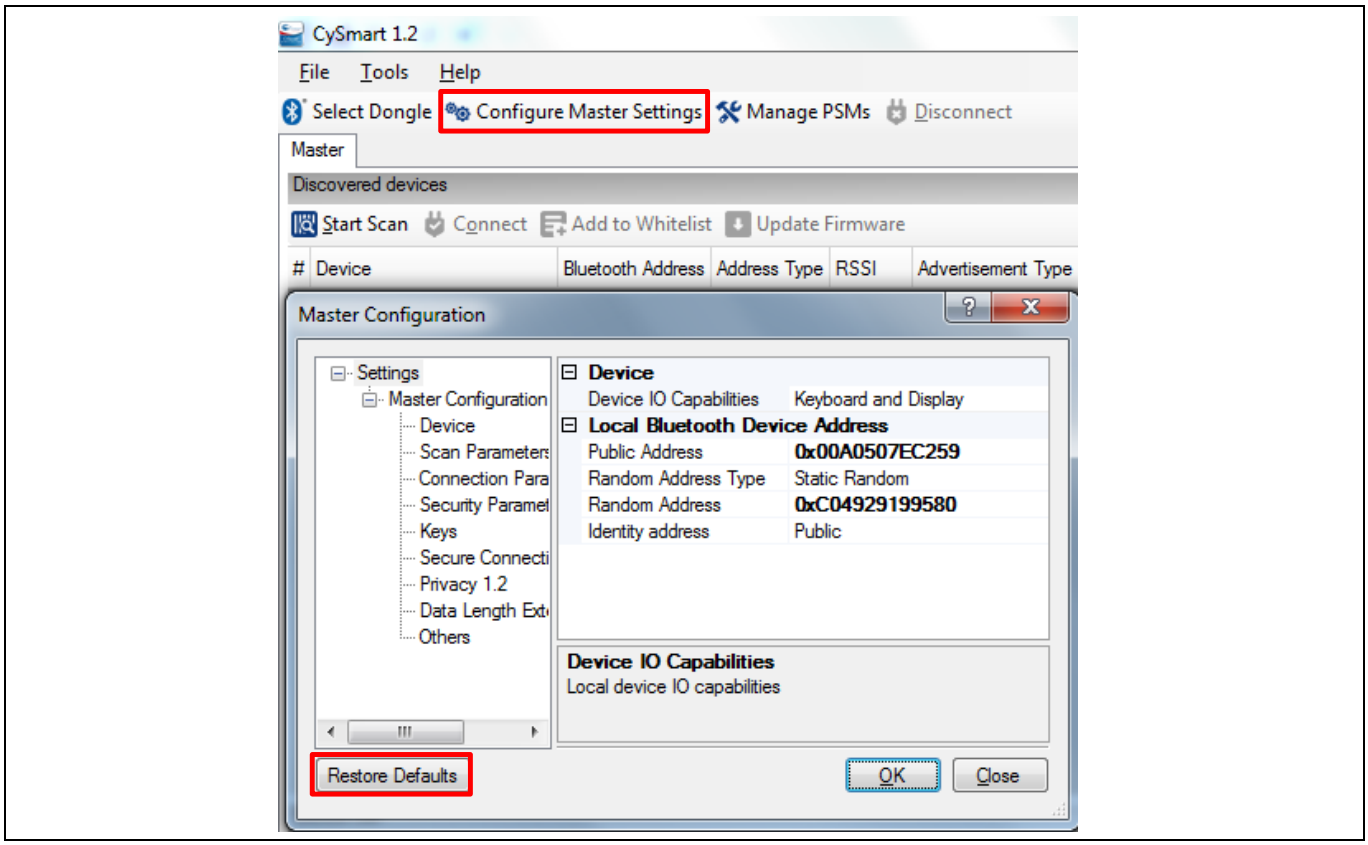


Figure 54 CySmart 主要设置的配置

4. 按下 Bluetooth® LE Pioneer 套件上的复位开关，启动您设计上的 Bluetooth® LE 广告。
5. 在 CySmart 主机仿真工具中，请点击 **Start Scan**。您的设备名称将显示在 **Discovered Devices** 列表中，如 **Figure 55** 所示。

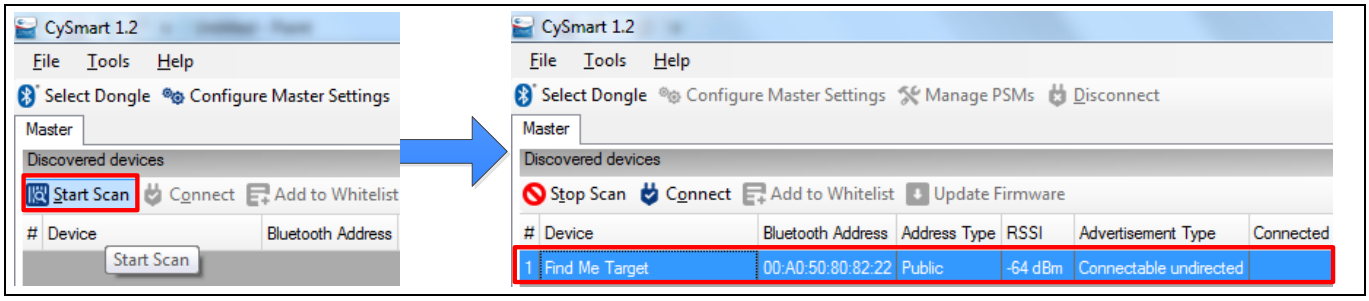


Figure 55 CySmart 设备检测

6. 选择您的设备并点击 **Connect**，从而建立 CySmart 主机仿真工具与您的设备间的 Bluetooth® LE 连接，如 **Figure 56** 所示。

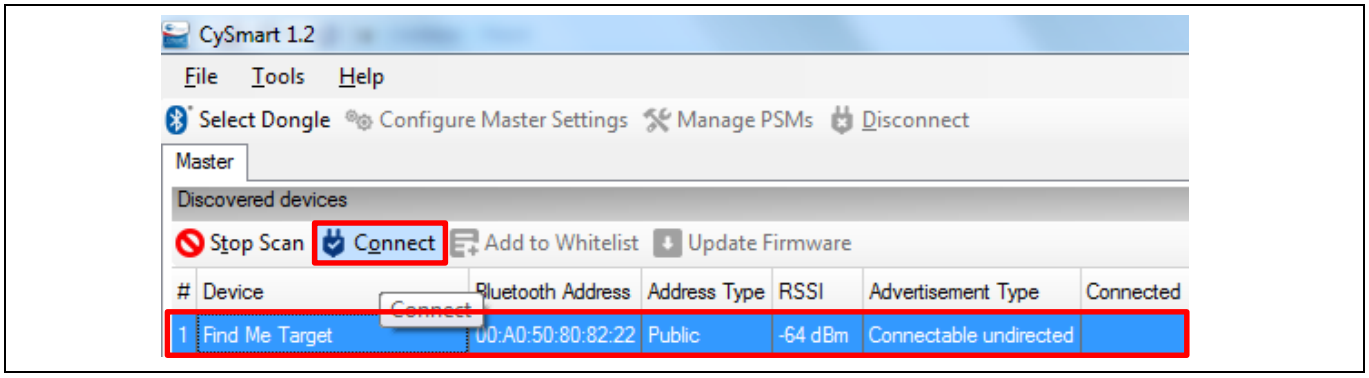


Figure 56 CySmart 设备连接

7. 建立好连接后，便可以从 CySmart 主机仿真工具中检测您的设计所具有的全部属性，如 **Figure 57** 所示。

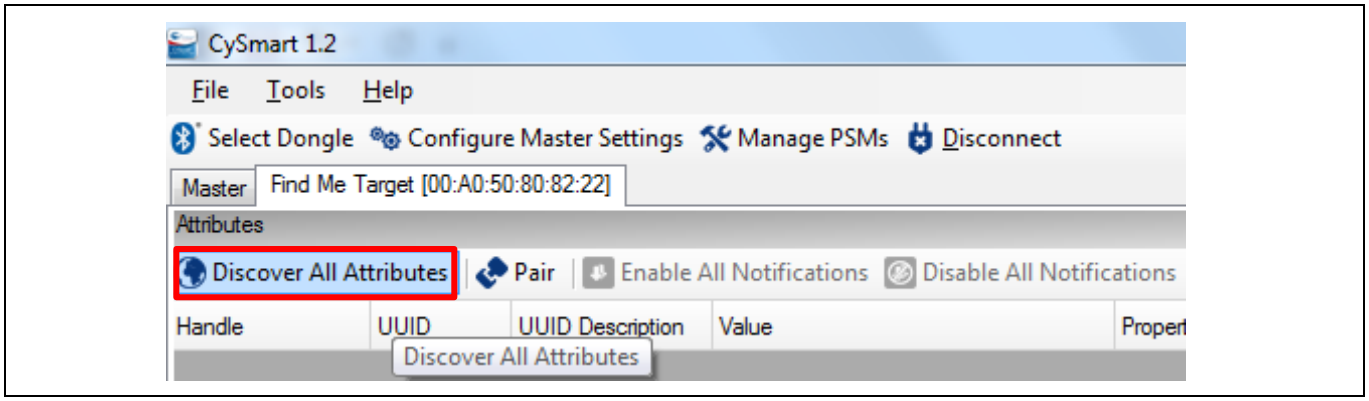


Figure 57 CySmart 属性检测

8. 在 Immediate Alert Service 下的 Alert Level Characteristic 项中输入数值 0、1 或 2，如 **Figure 58** 所示。观察您设备上的 LED 在不同的 Alert Level Characteristic 配置下的状态变化。

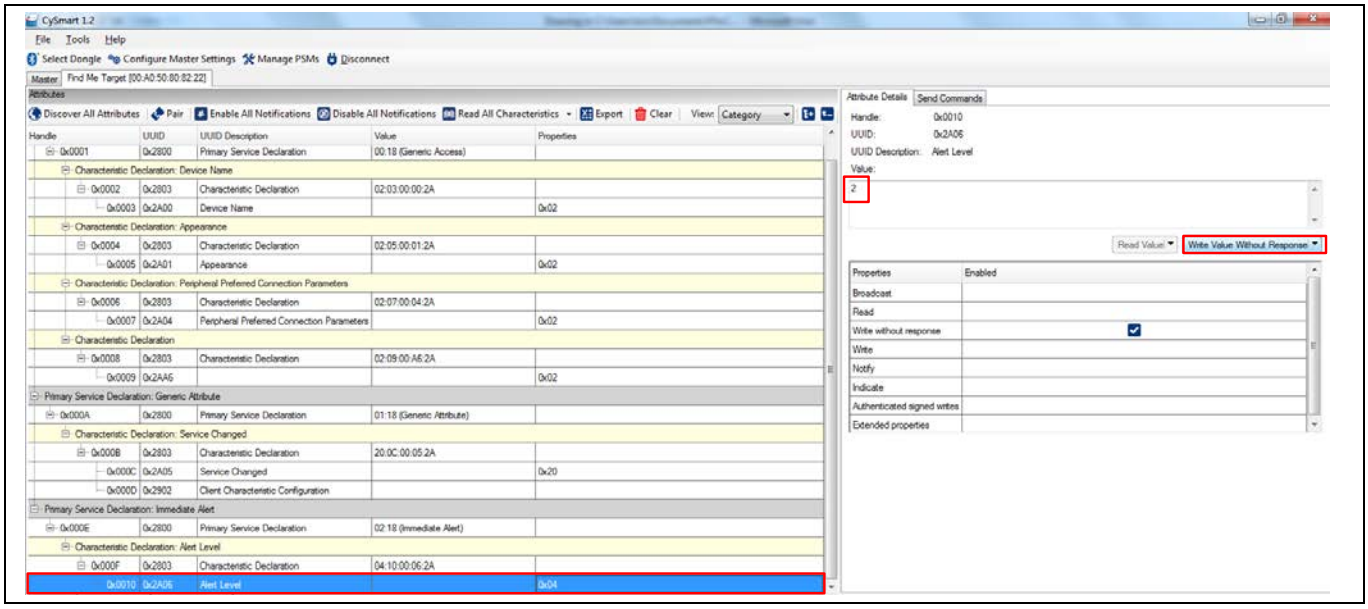


Figure 58　使用 CySmart 主机仿真工具进行测试

8 总结

本应用笔记介绍了 Bluetooth® LE 协议、PSoC™ 4 with AIROC™ Bluetooth® LE 器件架构和开发工具的基础知识。PSoC™ 4 with AIROC™ Bluetooth® LE 是真正的可编程嵌入式片上系统，在单芯片中集成了 Bluetooth® LE 射频、可配置的模拟和数字外设功能、存储器以及 Arm® Cortex®-M0 微控制器。由于集成了多项功能并支持各种低功耗模式，PSoC™ 4 with AIROC™ Bluetooth® LE 是可佩带电池供电、健康和运动 Bluetooth® LE 应用的理想选择。

此外，本应用笔记还向您提供了全部资源列表，用以加快深入对有关 PSoC™ 4 with AIROC™ Bluetooth® LE 知识的学习。

9 附录 A：Bluetooth® LE 器件系列的比较

Table 2 总结了 Infineon 的 Bluetooth® LE 器件系列的各项特性和功能。

Table 2 Bluetooth® LE 器件系列

特性	器件系列				
	CY8C41x7-BLxxx	**CY8C42x7-BLxxx**	**CYBL10xxx*** **	**CY8C41x8-BL**	**CY8C42x8-BL**
Bluetooth® LE 子系统	带有符合蓝牙 4.1 的协议栈的 Bluetooth® LE 射频和链路层硬件模块	带有符合蓝牙 4.1 的协议栈的 Bluetooth® LE 射频和链路层硬件模块	带有符合蓝牙 4.2 的协议栈的 Bluetooth® LE 射频和链路层硬件模块	带有符合蓝牙 4.2 的协议栈的 Bluetooth® LE 射频和链路层硬件模块**	带有符合蓝牙 4.2 的协议栈的 Bluetooth® LE 射频和链路层硬件模块**
蓝牙 4.2 特性	LE 安全连接	LE 安全连接	LE 安全连接、链路层保密以及链路层的数据长度扩展	LE 安全连接、链路层保密以及链路层的数据长度扩展**	LE 安全连接、链路层保密以及链路层的数据长度扩展**
CPU	带有单周期乘法的 24 MHz Arm® Cortex®-M0 CPU	带有单周期乘法的 48 MHz Arm® Cortex®-M0 CPU	带有单周期乘法的 48 MHz Arm® Cortex®-M0 CPU	带有单周期乘法的 24 MHz Arm® Cortex®-M0 CPU	带有单周期乘法的 48 MHz Arm® Cortex®-M0 CPU
闪存	128 KB	128 KB	256 KB	256 KB	256 KB
SRAM	16 KB	16 KB	32 KB	32 KB	32 KB
GPIO	最多 36 个	最多 36 个	最多 36 个	最多 36 个	最多 36 个
CapSense™	最多 35 个传感器	最多 35 个传感器	最多 35 个传感器	最多 35 个传感器	最多 35 个传感器
CapSense™手势	在选定的器件上	在选定的器件上	在选定的器件上	在选定的器件上	在选定的器件上
ADC	带有定序器的 12 位、806 ksps SAR ADC	带有定序器的 12 位、1 Msps SAR ADC	带有定序器的 12 位、1 Msps SAR ADC	带有定序器的 12 位、806 ksps SAR ADC	带有定序器的 12 位、1 Msps SAR ADC
运算放大器	两个可编程的运算放大器在深度睡眠模式下处于活动状态。	四个可编程的运算放大器在深度睡眠模式下处于活动状态。	无	两个可编程的运算放大器在深度睡眠模式下处于活动状态。	四个可编程的运算放大器在深度睡眠模式下处于活动状态。
比较器	两个支持唤醒功能的低功耗比较器	两个支持唤醒功能的低功耗比较器	无	两个支持唤醒功能的低功耗比较器	两个支持唤醒功能的低功耗比较器
电流 DAC	一个为 7 位，一个为 8 位	一个为 7 位，一个为 8 位	无	一个为 7 位，一个为 8 位	一个为 7 位，一个为 8 位
电源电压范围	1.9 V 至 5.5 V	1.9 V 至 5.5 V	1.9 V 至 5.5 V	1.9 V 至 5.5 V	1.9 V 至 5.5 V

附录 A：Bluetooth® LE 器件系列的比较

特性	器件系列				
	CY8C41x7-BLxxx	**CY8C42x7-BLxxx**	**CYBL10xxx*** **	**CY8C41x8-BL**	**CY8C42x8-BL**
低功耗模式	深度睡眠模式的电流为 1.3 µA 休眠模式的电流为 150 nA 停止模式的电流为 60 nA	深度睡眠模式的电流为 1.3 µA 休眠模式的电流为 150 nA 停止模式的电流为 60 nA	深度睡眠模式的电流为 1.3 µA 休眠模式的电流为 150 nA 停止模式的电流为 60 nA	深度睡眠模式的电流为 1.3 µA 休眠模式的电流为 150 nA 停止模式的电流为 60 nA	深度睡眠模式的电流为 1.3 µA 休眠模式的电流为 150 nA 停止模式的电流为 60 nA
段式 LCD 驱动	所选器件上的 4 条公用线、32 段式 LCD 驱动	所选器件上的 4 条公用线、32 段式 LCD 驱动	所选器件上的 4 条公用线、32 段式 LCD 驱动	所选器件上的 4 条公用线、32 段式 LCD 驱动	所选器件上的 4 条公用线、32 段式 LCD 驱动
串行通信	两个独立的串行通信模块 (SCB)，可将其配置为 I^2C、SPI 或 UART	两个独立的串行通信模块 (SCB)，可将其配置为 I^2C、SPI 或 UART	两个独立的串行通信模块 (SCB)，可将其配置为 I^2C、SPI 或 UART	两个独立的串行通信模块 (SCB)，可将其配置为 I^2C、SPI 或 UART	两个独立的串行通信模块 (SCB)，可将其配置为 I^2C、SPI 或 UART
定时计数脉宽调制器 (TCPWM)	4	4	4	4	4
通用数字模块 (UDB)	无	具有 4 个 UDB，每个 UDB 包括 8 个宏单元和一个数据路径。可以将其用于合成额外的数字外设 (定时器、计数器、脉冲宽度调制器 (PWM)) 或通信接口 (UART、SPI) 。	无	无	具有 4 个 UDB，每个 UDB 包括 8 个宏单元和一个数据路径。可以将其用于合成额外的数字外设 (定时器、计数器、脉冲宽度调制器 (PWM)) 或通信接口 (UART、SPI) 。
其他的数字外设 (I^2S、PWM)	无	有 (所选器件上的基于 UDB 的数字外设)	有 (所选器件上的固定功能模块)	无	有 (所选器件上的基于 UDB 的数字外设)
时钟	IMO 频率为 3 ~ 24 MHz ILO 频率为 32 kHz ECO 频率为 24 MHz WCO 频率为 32 kHz	IMO 频率为 3 ~ 48 MHz ILO 频率为 32 kHz ECO 频率为 24 MHz WCO 频率为 32 kHz	IMO 频率为 3 ~ 48 MHz ILO 频率为 32 kHz ECO 频率为 24 MHz WCO 频率为 32 kHz	IMO 频率为 3 ~ 24 MHz ILO 频率为 32 kHz ECO 频率为 24 MHz WCO 频率为 32 kHz	IMO 频率为 3 ~ 48 MHz ILO 频率为 32 kHz ECO 频率为 24 MHz WCO 频率为 32 kHz
电源监控	上电复位 (POR) 欠压检测 (BOD) 低压检测 (LVD)	POR BOD LVD	POR BOD LVD	POR BOD LVD	POR BOD LVD

附录 A：Bluetooth® LE 器件系列的比较

特性	器件系列				
	CY8C41x7-BLxxx	**CY8C42x7-BLxxx**	**CYBL10xxx*** **	**CY8C41x8-BL**	**CY8C42x8-BL**
封装	56-QFN (7.0 × 7.0 × 0.6 mm)，68-WLCSP (3.52 × 3.91 × 0.55 mm)	56-QFN (7.0 × 7.0 × 0.6 mm)，68-WLCSP (3.52 × 3.91 × 0.55 mm)	56-QFN* (7.0 × 7.0 × 0.6 mm)，76-WLCSP (4.04 × 3.87 × 0.55 mm)	56-QFN* (7.0 × 7.0 × 0.6 mm)，76-WLCSP (4.04 × 3.87 × 0.55 mm)	56-QFN* (7.0 × 7.0 × 0.6 mm)，76-WLCSP (4.04 × 3.87 × 0.55 mm)
DMA	无	无	最多 8 个通道	最多 8 个通道**	最多 8 个通道**

* = CY8C41x8-BL 和 CY8C42x8-BL (256K 闪存) QFN 封装的引脚分别与 CY8C41x7-BL 和 CY8C42x7-BL (128K 闪存) QFN 封装的引脚兼容。

** = CY8C41x8-BL 和 CY8C42x8-BL 系列都具有两个小系列 (–BL4xx 和–BL5xx)。–BL4xx 系列不支持蓝牙 4.2 链路层保密、链路层的数据长度扩展和 DMA 等功能；但–BL5xx 系列则支持这些特性。

*** = CY8CBL10xxx PRoC 器件系列具有多个小系列，此处所描述的特性是针对超集器件的。请参考 **AN94020 - PRoC™ BLE 入门**，了解详情。

10 附录 B：Infineon 专业术语

本部分列出了在操作 Infineon PSoC™器件系列时可能遇到的常用术语。

组件配置工具：是指每个组件中嵌入的 PSoC™ Creator 的简单图形用户界面 (GUI)。通过该工具，您可以自定义组件参数，并且可以通过右键点击某个组件来访问它。

组件：由 PSoC™ Creator 软件中的一个图标表示的免费嵌入式芯片。它们用于将多个芯片和系统接口集成到已通过主系统总线与 MCU 连接的 PSoC™组件上。例如，通过 Bluetooth® LE 组件，可以在几分钟内构建 Bluetooth Smart 设备。同样，您可以将可编程模拟组件作为传感器使用。

MiniProg3：是指用于进行开发的硬件，即在不支持内置编程器的自定义电路板或 PSoC™开发套件上编程 PSoC™器件。

PSoC™：PSoC™是一个包含 CPU (如 32 位 Arm® Cortex®-M0)、可编程的模拟和数字模块的可编程嵌入式设计平台。通过它，可以使用稳定且易于使用的解决方案 (如触摸感应) 来加快嵌入式系统的设计，并能够创建低功耗设计。

PSoC™ 4 with AIROC™ Bluetooth® LE：它是一个集成了 Bluetooth® LE 射频的 PSoC™ 4 芯片。该 Bluetooth® LE 射频模块包括一个符合蓝牙 4.2 规范的买断式授权 Bluetooth® LE 协议栈。

PSoC™ Creator：是 PSoC™ 3、PSoC™ 4 和 PSoC™ 5LP 的集成开发环境 (IDE) 软件，该软件安装在电脑上，通过使用它可同时设计 PSoC™系统的硬件和固件，或允许设计硬件，然后将这些设计移植给其他流行 IDE 软件。

PSoC™ Programmer：是用于编程 PSoC™器件的灵活的集成式编程应用程序。PSoC™ Programmer 和 PSoC™ Creator 用于对 PSoC™ 3、PSoC™ 4、PRoC 和 PSoC™ 5LP 设计进行编程。

11 附录 C：Infineon Bluetooth® LE 开发工具

PSoC™ 4 with AIROC™ Bluetooth® LE Pioneer 开发工具

Figure 59 中显示的 Bluetooth® LE Pioneer 套件是 Infineon 的 Bluetooth® LE 开发工具，它支持 PSoC™ 4 with AIROC™ Bluetooth® LE 器件系列。该套件还包括连接到一个 Bluetooth® LE Pioneer 基板的可插入 PSoC™ 4 with AIROC™ Bluetooth® LE (和 PRoC-BLE) 模块。该套件由一个纽扣电池供电或通过 USB 接口供电。

Bluetooth® LE Pioneer 基板和 Bluetooth® LE 模块的组合有助于开发电池供电的低功耗 Bluetooth® LE 设计，该 Bluetooth® LE 结合与标准的 Arduino Uno 连接器相兼容的屏蔽或板上 PSoC™ 4 with AIROC™ Bluetooth® LE 器件功能 (如 CapSense™用户界面) 一起使用。

Bluetooth® LE Pioneer 基板带有一个板上编程和调试接口，使 Bluetooth® LE 设计通过 PSoC™ Creator IDE 能够快速且容易地进行调试。Bluetooth® LE Pioneer 套件还带有一个插座，支持与第三方供应商 (如 Digilent®) 的 Pmod™子卡连接。该套件还包括一个作为 Bluetooth® LE 连接主设备使用的 Bluetooth® LE 收发器，该收发器结合与 CySmart 主机仿真工具一起使用，这样能够为非 Bluetooth® LE Windows 电脑系统提供 Bluetooth® LE 主机仿真平台。

它还包括一组 Bluetooth® LE 示例项目和一系列文档，可助您开始开发自己的 Bluetooth® LE 应用。请访问 **Bluetooth® LE Pioneer 套件**网址，以获得该套件的最新版本，并下载套件设计、示例项目以及文档文件。

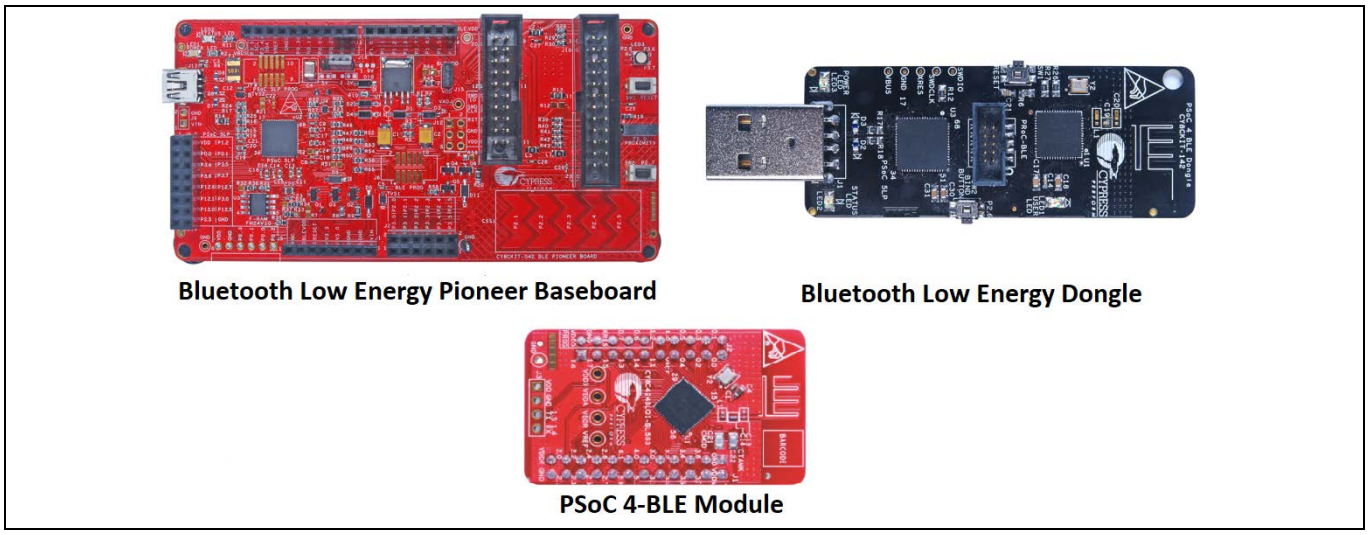


Figure 59 Bluetooth® LE Pioneer 套件

11.1 CySmart 主机仿真工具

CySmart 主机仿真工具是一个 Windows 应用，它使用 Bluetooth® LE Pioneer 套件的 Bluetooth® LE 收发器进行仿真 Bluetooth® LE 中心器件；如 Figure 19 所示。将它作为 Bluetooth® LE Pioneer 套件的一部分进行安装，并且可以通过右击 Bluetooth® LE 组件中的选项来启动它。它还提供了一个平台，这样您可以寻找和配置外设上的 Bluetooth® LE 服务、特性和属性，从而在 GATT 或 L2CAP 面向连接通道上可以测试 PSoC™ 4 with AIROC™ Bluetooth® LE 外设的实现。

通过 CySmart 主机仿真工具，您可以执行 (并不限于) 以下操作：

- 扫描 Bluetooth® LE 外设，以寻找可连接的有效器件。
- 发现可用的 Bluetooth® LE 属性，包括连接外设器件的服务和特性。
- 对特性值和描述符进行读/写操作。
- 接收来自所连接外设器件的特性通知和指示。

附录 C：Infineon Bluetooth® LE 开发工具

- 通过使用 Bluetooth® LE 安全管理器程序进行贴合所连接的外设器件。
- 为外设器件建立 Bluetooth® LE L2CAP 面向连接的会话，并根据蓝牙 4.2 规范进行切换数据。
- Infineon Bluetooth® LE 外设器件的空中传送 (OTA) 固件更新

Figure 60 和 **Figure 61** 显示的是 CySmart 主机仿真工具的用户界面。有关设置和使用该工具的详细信息，请参见**帮助**菜单中的 CySmart 用户指南。

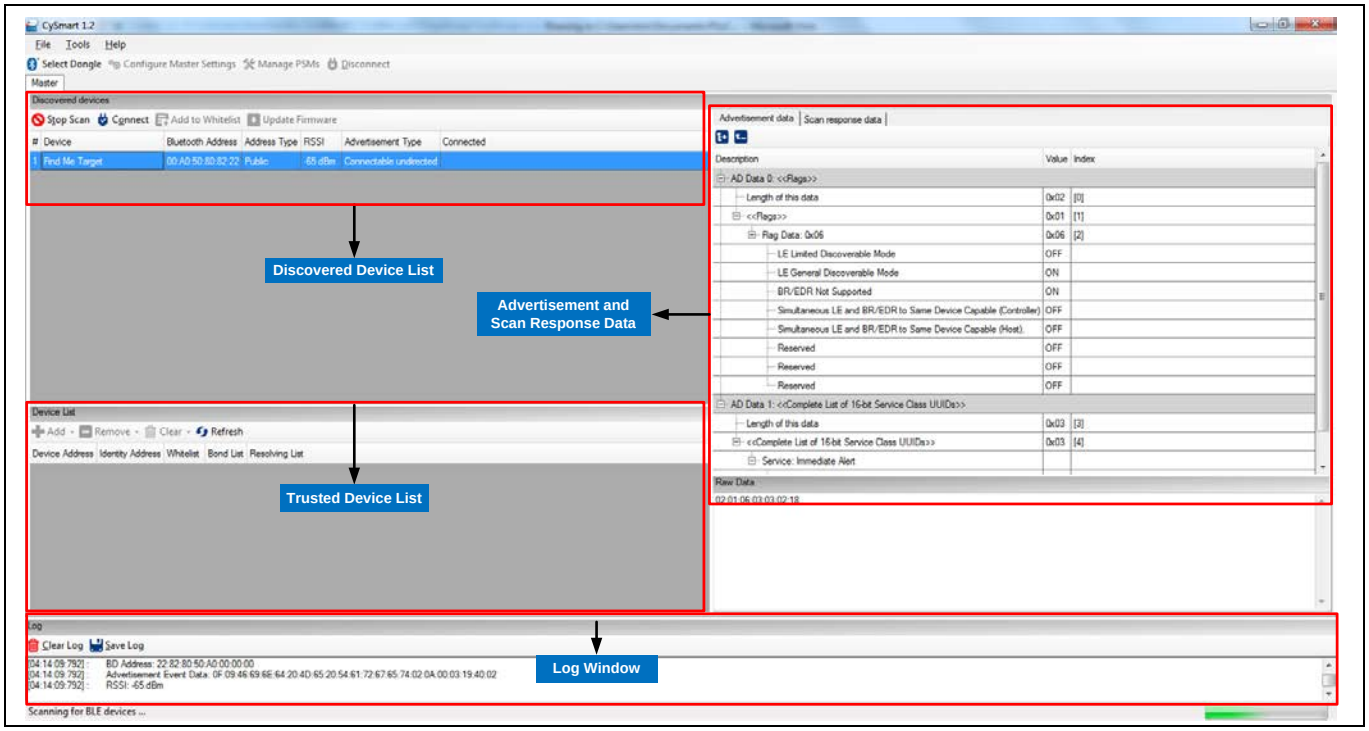


Figure 60　CySmart 主机仿真工具主设备选项卡

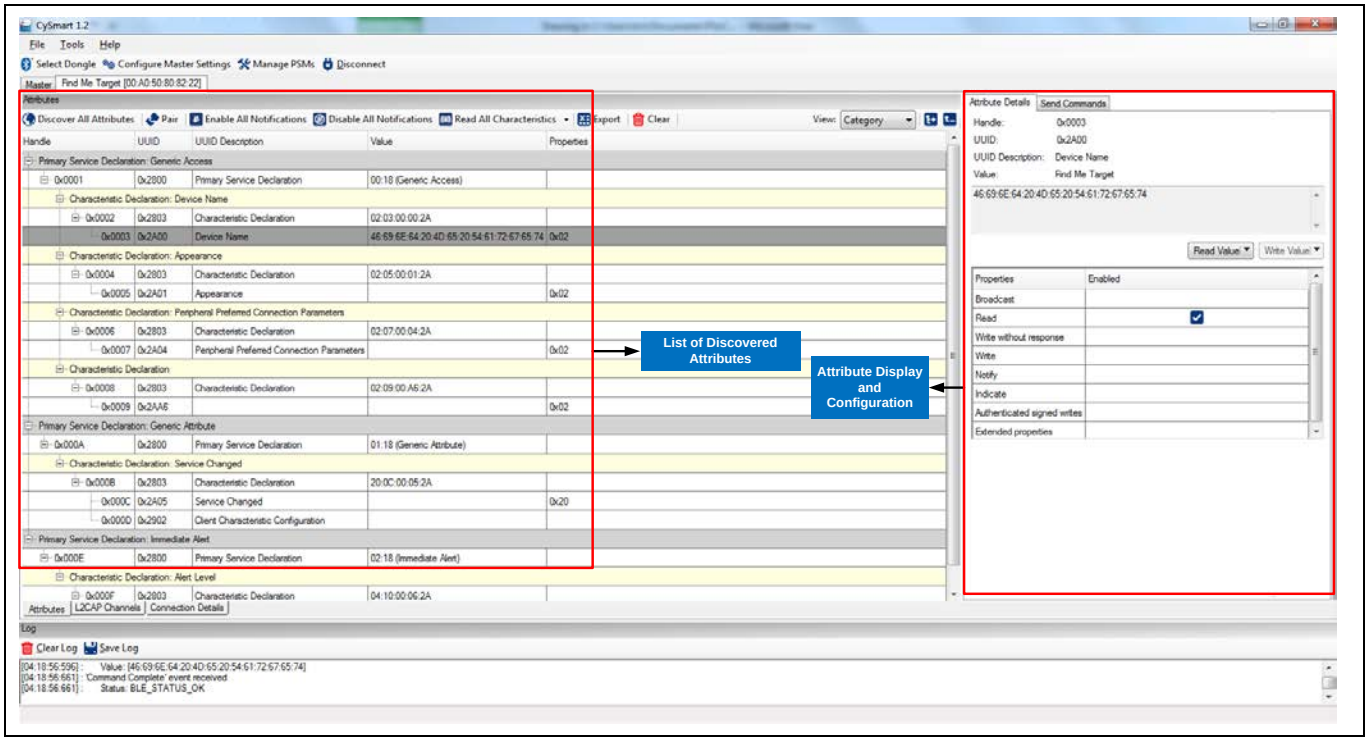


Figure 61　CySmart 主机仿真工具外设器件属性选项卡

11.2　手机版的 CySmart 应用

除电脑工具外，您还可以从 iOS 或 Android 版本的应用程序商店中下载相应的手机版的 CySmart 应用。该应用分别为 Bluetooth® LE 使用了 iOS 内核蓝牙框架和 Android 内置平台框架，以便将 Bluetooth® LE 使能的智能手机配置为可扫描器件，并能够连接外设器件的中心设备。

手机应用通过直观的 GUI 支持 SIG 采用的 Bluetooth® LE 标准配置文件，并总结了基础的 Bluetooth® LE 服务和特性细节。除了 Bluetooth® LE 标准配置外，该应用还使用 Infineon LED 和 CapSense™演示示例来显示自定义配置文件的实现。**Figure 62** 和 **Figure 63** 显示的是一组心率配置文件用户界面的 CySmart 应用屏幕截图。有关如何使用具有 Bluetooth® LE Pioneer 套件示例项的应用的详细信息，请参见 Bluetooth® LE Pioneer 套件指南。

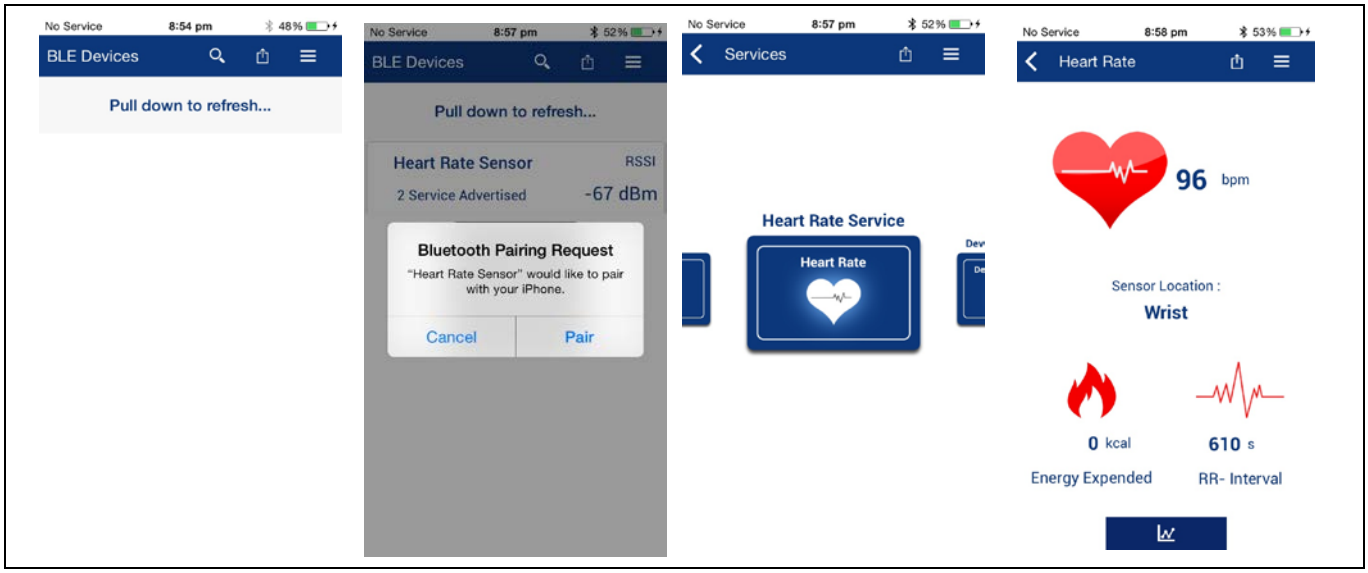


Figure 62　使用 iOS 版本的 CySmart 应用的心率配置文件示例

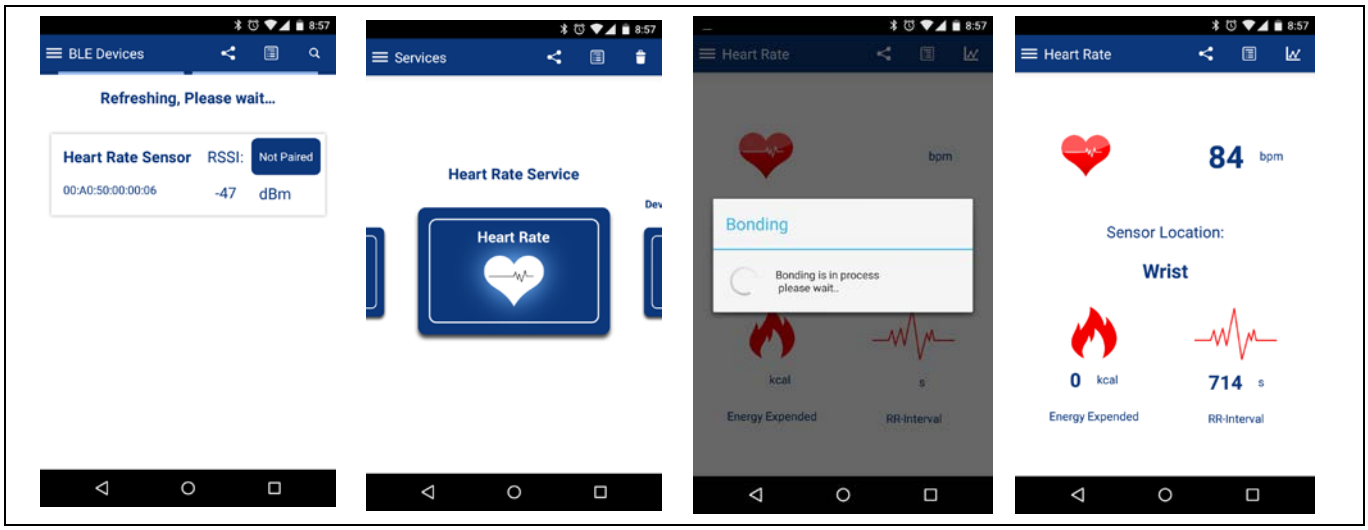


Figure 63　使用 Android 版本的 CySmart 应用的心率配置文件示例

12 附录 D：PSoC™ 4 with AIROC™ Bluetooth® LE 器件

12.1 Bluetooth® Low Energy 子系统 (BLESS)

Bluetooth® LE 子系统包括物理层 (PHY) 、和带有嵌入式 AES-128 安全性引擎的链路层。

物理层包含符合蓝牙 4.2 规范的数字 PHY 和 RF 收发器。收发器以 1 Mbps 的速率通过 2.4 GHz ISM 频带发送和接收 GFSK 数据包。基带控制器是一种复合的硬件/固件实现系统，它支持主设备模式和从设备模式。关键协议因素 (如 HCI 和链路控制) 都是通过固件实现，而实时要求严格的功能 (如加密、CRC、数据白化处理以及访问代码相关性) 则是通过硬件实现。

BLESS 与蓝牙 4.2 规范相兼容，它不仅支持蓝牙 4.0 规范中的所有功能，并且还支持 4.2 规范中的几项附加性能，如低占空比广播、LE ping 服务、L2CAP 面向连接的通道、链路层保密、链路层的数据长度扩展以及 LE 安全连接。BLESS 模块还包含一个外部晶振 (ECO) 和一个时钟晶振 (WCO) ，分别用于生成准确的 RF 频率并保持 Bluetooth® LE 链路上持续连接时间间隔的时长。BLESS 模块中的 ECO 和 WCO 可作为 PSoC™ 4 with AIROC™ Bluetooth® LE 系统的高频和低频时钟源使用。请参见时钟系统。

BLESS 支持五种功能功耗模式，分别为深度睡眠、睡眠、空闲、发送和接收模式。

Note: *本节介绍的这些功耗模式都是针对 BLESS 模块的。有关 PSoC™ 4 with AIROC™ Bluetooth® LE 系统的功耗模式，请参见低-漏功耗模式部分。*

12.1.1 深度睡眠模式

深度睡眠模式是 BLESS 支持的最低功耗模式。在该模式下，射频功能被关闭。在发送和接收数据包完成后的广播或连接间隔期间将进入该模式，这样可以尽量节省电源。为了节省电源，可在该模式下关闭 ECO。但要保持 WCO (即低频时钟) 为打开状态，以维持 Bluetooth® LE 链路层的时序参考逻辑。进入和退出该模式是由 CPU 控制的。

12.1.2 睡眠模式

在睡眠模式下，射频功能被关闭。BLESS 模块保持了所有配置内容。ECO 和 WCO 均被打开，但内核 BLESS 逻辑的时钟被关闭。进入和退出该模式是由 CPU 控制的。

12.1.3 空闲模式

空闲模式是发送和接收模式的就绪状态。在该模式下，射频功能被关闭，但链路层逻辑的链路层时钟保持为启用状态，这样可以使 CPU 启动协议状态机。

12.1.4 发送模式

发送模式是一种活动模式。在该模式下，BLESS 中的所有模块将被通电。链路层时钟被启用，以完成链路层和 RF-PHY 间的逻辑运算。在该模式下，RF-PHY 接收来自链路层的速率为 1 Mbps 的串行数据，并将 2.4 GFSK 调制的数据发送到天线端口上。这时，Bluetooth® LE 会从空闲模式转入到发送模式。

12.1.5 接收模式

BLESS 进入接收模式，以执行 Bluetooth® LE 的特定接收器操作。RF-PHY 对来自 RF 模拟模块的 1 Mbps 数据进行解析，并在解调后将其转发给链路层控制器。有关 BLESS 功耗模式和工作子模块的总结，请参见 **Table 3**。

Table 3 BLESS 功耗模式

BLESS 功耗模式	ECO	WCO	RF Tx	RF Rx	BLESS 内核
深度睡眠	关闭	开启	关闭	关闭	关闭
睡眠	开启	开启	关闭	关闭	关闭
空闲	开启	开启	关闭	关闭	开启
发送	开启	开启	开启	关闭	开启
接收	开启	开启	关闭	开启	开启

12.2 Arm® Cortex®-M0、存储器和 DMA

PSoC™ 4 with AIROC™ Bluetooth® LE 具有一个基于 32 位 Arm® Cortex®-M0 的 CPU，其最大工作频率为 48 MHz，提供了 43-DMIPS 性能。CPU 支持单周期 32 位乘法。

PSoC™ 4 with AIROC™ Bluetooth® LE 具有 16 KB/32 KB 大小的 SRAM 以及包含一个读取加速器的 128 KB 大小的闪存，因此它能够提供通用的 Bluetooth® LE 应用。该器件也提供了一个 512 字节的监控闪存区，用于存储用户特定的数据，如 Bluetooth® LE 器件地址和加密密钥。

通过 PSoC™ 4XX8-BL 器件系列所提供的 8 通道 DMA 传输，可以在外设与外设、存储器与存储器或外设与存储器之间执行 32 位的数据传输操作。

12.3 可编程的数字外设

PSoC™ 4 with AIROC™ Bluetooth® LE 提供了一组丰富的数字外设，包括：可编程串行通信模块 (SCB) 、定时计数脉宽调制器 (TCPWM) 以及被称为通用数字模块 (UDB) 的可编程逻辑阵列。

12.3.1 可编程 SCB

PSoC™ 4 with AIROC™ Bluetooth® LE 含有独立运行的 SCB，可将其配置为 I^2C、SPI 或 UART。SCB 支持以下各项特性：

- 标准 SPI 主从设备功能，支持 Motorola®、Texas Instruments®和 National Semiconductor®的协议
- 标准 UART 功能，支持智能卡阅读器、本地互联网络 (LIN) 以及红外数据组织 (IrDA) 协议
- 标准的 I^2C 主从设备功能
- SPI 和 EZI^2C 模式 (其允许器件正常操作而无需 CPU 的干预)
- 针对 SPI 和 I^2C 协议的低功耗 (深度睡眠) 模式 (使用外部时钟)

有关详细信息，请参考 **PSoC™ 4 SCB 组件的数据手册**。

12.3.2 可编程 TCPWM

PSoC™ 4 with AIROC™ Bluetooth® LE 具有四个 16 位的可编程 TCPWM 模块。每个 TCPWM 可以实现 16 位定时器、计数器、PWM 或正交解码器功能。TCPWM 提供了互补输出以及可选的启动、重载、停止、计数和捕获事件等信号。PWM 模式支持中心对齐、边沿对齐和伪随机等操作。

有关详细信息，请参考 **PSoC™ 4 TCPWM 组件的数据手册**。

12.3.3 通用数字模块

UDB 是可编程的逻辑模块，它提供了同 CPLD 和 FPGA 模块相似的各项功能，如 **Figure 64** 所示。通过使用 UDB，可以创建多种数字功能，如定时器、计数器、PWM、伪随机序列 (PRS) 、CRC、移位寄存器、SPI、UART、I2S，以及自定义的组合和连续逻辑电路。

每个 UDB 包含两个可编程逻辑组件 (PLD)，每个组件含有 12 个输入和 8 个乘积项。PLD 能够构成锁存或组合“乘积和”逻辑。此外，每个 UDB 还包括一个 8 位的单周期算术逻辑单元 (ALU)，该单元被称为 “Datapath” (数据路径)。通过该数据路径，可以有效实现各项功能，如定时器、计数器、PWM 和 CRC。

UDB 同样也提供了一个交换式数字信号互连 (DSI) 结构，能够将外设和端口中的信号路由到 UDB 或通过 UDB 进行通信和控制。

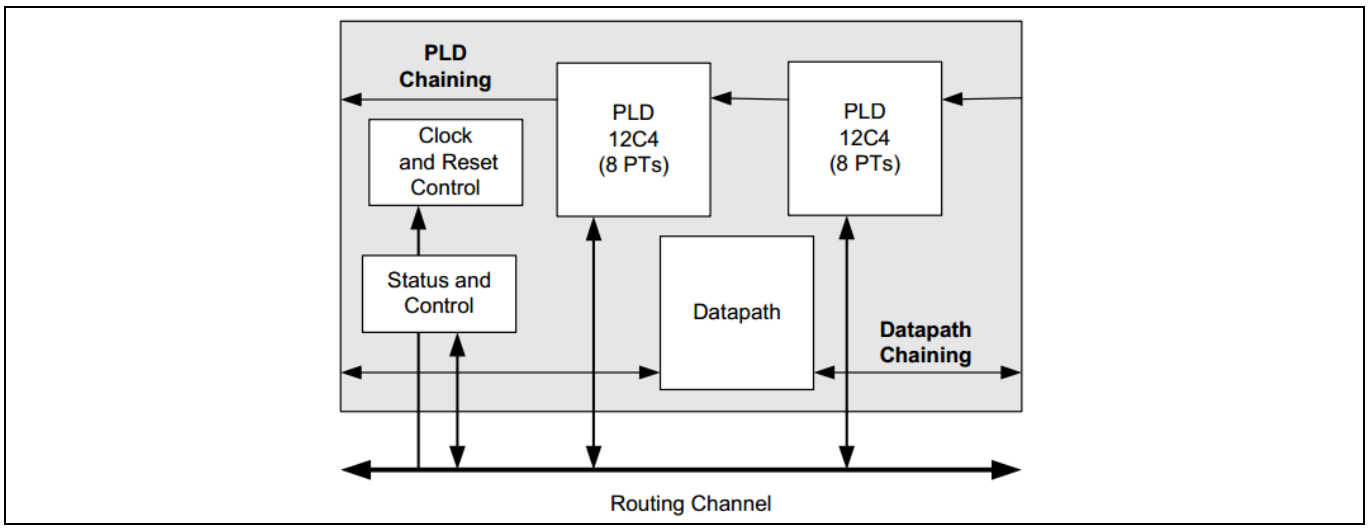


Figure 64　通用数字模块框图

您无需掌握任何硬件描述语言 (HDL)，但仍能使用 UDB。PSoC™ Creator (Infineon PSoC™ 4 with AIROC™ Bluetooth® LE 的开发工具) 能够根据原理图生成所需功能。需要时，高级用户可以使用 Verilog 实现 UDB 上的自定义逻辑。

更多相关信息，请参见下面的应用笔记。

- **AN62510 — 使用 PSoC™ 3、PSoC™ 4 和 PSoC™ 5LP 实现状态机功能**
- **AN82156 — PSoC™ 3、PSoC™ 4 和 PSoC™ 5LP —根据 UDB 数据路径进行设计 PSoC™ Creator 组件**
- **AN82250 — PSoC™ 3、PSoC™ 4 和 PSoC™ 5LP —使用 Verilog 实现可编程逻辑设计**

12.3.4　应用

Table 4 显示的是 Bluetooth® LE 应用中可编程数字外设的使用情况。

Table 4　可编程数字外设的应用

应用	数字外设
传感器集线器	I^2C (数字传感器接口)、PWM (执行器)、I2S (语音输入)
健康和运动	PWM (用户界面)、计数器 (波形峰值测量)、SPI (外部存储器接口)
工业级	UART (modbus)、计数器 (事件计数)
家庭自动化	PWM (车库门控制)、PWM (灯光控制)

12.4　可编程模拟模块

PSoC™ 4 with AIROC™ Bluetooth® LE 提供了业界一流的模拟集成系统。模拟系统包括微型的连续时间模块 (CTBm)、一个高速的 12 位 SAR ADC、低功耗比较器、电容式触摸感应 (CapSense™) 以及段式 LCD 直接驱动程序。

12.4.1 微型的连续时间模块 (CTBm)

PSoC™ 4 with AIROC™ Bluetooth® LE 具有两个 CTBm 模块；其中每个模块包括两个可编程运算放大器和一个开关矩阵。您可以单独将每个运算放大器配置为比较器、电压跟随器或具有外部反馈性能的运算放大器。需要时，可以配置 CTBm 模块，从而可以运行于深度睡眠模式。

有关详细信息，请参考下面的组件数据手册：

- **PSoC™ 4 运算放大器**
- **PSoC™ 4 电压比较器**

12.4.2 带有硬件定序器的 SAR ADC

PSoC™ 4 with AIROC™ Bluetooth® LE 具有带输入通道的 1 Msps、12 位逐次逼近寄存器 (SAR) ADC，这些输入通道支持可编程的分辨率以及单端或差分输入选项。GPIO 的数量限制了可能被实现的 ADC 输入通道的数量。

SAR ADC 具有一个硬件定序器，允许在 8 个通道上执行自动扫描操作而无需 CPU 的干预。它还支持预处理操作，比如对 8 个通道上的输出数据进行累加和求平均操作。

您可使用固件、定时器、引脚或 UDB 等方法来触发扫描操作，从而使设计更加灵活。

有关详细信息，请参考 **PSoC™ 4 SAR ADC 组件的数据手册**。

12.4.3 低功耗比较器

PSoC™ 4 with AIROC™ Bluetooth® LE 器件包含多个低功耗比较器，它们能够在所有功耗模式 (**停止**模式除外) 下运行。在比较注重功耗的设计中，当器件进入低功耗模式时，您可以使用低功耗比较器来监控模拟输入，并生成一个可唤醒系统的中断。

更多有关信息，请参考 **PSoC™ 4 低功耗比较器组件数据手册**。

12.4.4 电容式触摸感应 (CapSense™)

通过检测人体的电容，电容触摸传感器可以检测在传感器上或者附近是否存在手指。电容传感器外观漂亮，简单易用，且使用寿命长。

PSoC™ 4 with AIROC™ Bluetooth® LE 中的 CapSense™特性可提供前所未有的信噪比、一流的防水功能，以及各种传感器类型，如按键、滑条、触控板和接近传感器。

通过 Infineon 所提供的软件组件，可以轻松创建电容感应设计；组件支持名称为 SmartSense™的自动硬件调校功能，并为触控板和接近传感器提供了手势识别库。

如果不使用电容感应，那么可以将 CapSense™模块中两个电流 DAC (IDAC) (一个为 7 位和一个为 8 位) 作为通用 IDAC 使用。可将 CapSense™模块中的比较器作为通用比较器使用。

更多有关信息，请参考 **PSoC™ 4 CapSense™设计指南**。

12.4.5 段式 LCD 直接驱动程序

大部分低功耗、便携式的手持设备 (如血糖仪、万用表和血压计) 都使用了段式 LCD 来显示信息。段式 LCD 通常需要通过外部驱动器连接到微控制器。PSoC™ 4 with AIROC™ Bluetooth® LE 包含了一个集成的低功耗 LCD 驱动程序，它可以直接驱动段式 LCD 显示屏。

PSoC™ 4 with AIROC™ Bluetooth® LE 可驱动具有 4 COM x 32 SEG 电极的 LCD。段式 LCD 驱动器可在深度睡眠下保持显示屏内容，此时系统的电流消耗仅为 7 µA。

更多相关信息，请参考 **AN87391 — PSoC™ 4 的段式 LCD 直接驱动功能**。

12.4.6 应用

Table 5 列出了不同 Bluetooth® LE 应用中可编程模拟外设的使用情况。

Table 5 可编程模拟外设的应用

应用	模拟外设
健康和运动	运算放大器：TIA (心率测量) 运算放大器：跟随器 (模拟-参考缓冲器) ADC (采样心率信号) CapSense™ (腕套上的用户界面) 段式 LCD (腕套上的显示屏)
传感器集线器	ADC (模拟传感器接口) 模拟复用器 (多传感器输入) 运算放大器 (信号放大器) 段式 LCD (UI)
工业级	ADC：差分模式 (温度测量) IDAC (温度传感器驱动) 低功耗比较器 (检测到阈值时会唤醒器件) 运算放大器：PGA (4 mA 到 20 mA 电流回路系统)
家庭自动化	运算放大器：PGA (运动传感器，光线传感器) 比较器 (门锁检测器) ADC 和运算放大器滤波器 (烟雾探测器)

12.5 系统资源

本节说明了 PSoC™ 4 with AIROC™ Bluetooth® LE 中所有外设可用的系统资源。

12.5.1 低-漏功耗模式

PSoC™ 4 with AIROC™ Bluetooth® LE 提供了下面各种功耗模式。请注意，这是 PSoC™ 4 with AIROC™ Bluetooth® LE 器件的功耗模式，它们与 **Bluetooth® Low Energy 子系统 (BLESS)** 一节所说明的功耗模式不同。

- **活动模式**：它是器件的主要工作模式。在该模式下，所有外设都可用。
- **睡眠模式**：在该模式下，CPU 处于睡眠模式，SRAM 数据保持不变，并且所有外设均可用。任何中断都能唤醒 CPU，并使系统返回到活动模式。
- **深度睡眠模式**：在该模式下，高频率时钟 (IMO) 和所有高速外设都被关闭。而低频率时钟 (32 kHz ILO 和 WCO) 和慢速外设均可用。通过低速、异步或低功耗模拟外设的中断，可唤醒器件。在该模式下，电流消耗为 1.3 µA。
- **休眠模式**：该功耗模式提供了最佳电流消耗 (150 nA)，但仍能保持 SRAM 数据以及响应由低功耗比较器或 GPIO 生成的中断唤醒。
- **停止模式**：该功耗模式保留了 GPIO 的状态。唤醒事件可以通过固定的 WAKEUP 引脚触发。在该模式下，仅消耗了 60 nA 的电流。

在电池供电的 Bluetooth® LE 系统中，您可以结合使用睡眠、深度睡眠、休眠和停止模式，以达到最佳的系统电源，使电池寿命更长。

附录 D：PSoC™ 4 with AIROC™ Bluetooth® LE 器件

Table 6 显示的是 PSoC™ 4 with AIROC™ Bluetooth® LE 系统功耗模式与 BLESS 功耗模式间的依赖关系。所有这些依赖关系可通过简单的 API 自动处理；请参考主循环和低功耗操作一节中的示例。

在一个典型的 Bluetooth® LE 应用 (如心率监测) 中，PSoC™ 4 with AIROC™ Bluetooth® LE 器件测量心率时会处于活动模式；Bluetooth® LE 射频在发送或接收数据包时该器件会处于睡眠模式；在连续 Bluetooth® LE 连接的间隔至之间，该器件会处于深度睡眠模式，并在 Bluetooth® LE 广播超时时它会处于休眠或停止模式。

Table 6 PSoC™ 4 with AIROC™ Bluetooth® LE 功耗模式

BLESS 模式	PSoC™ 4 with AIROC™ Bluetooth® LE 系统功耗模式				
	活动	睡眠	深度睡眠	休眠	停止
发送	✓	✓	×	×	×
接收	✓	✓	×	×	×
空闲	✓	✓	×	×	×
睡眠	✓	✓	×	×	×
深度睡眠	✓	✓	✓	×	×
供电	×	×	×	✓	✓

12.5.2 电源供应及监测

PSoC™ 4 with AIROC™ Bluetooth® LE 可以在 1.9 V 到 5.5 V 的电源范围内运行。可以使用多个内部电压调节器来支持器件的不同功耗模式。PSoC™ 4 with AIROC™ Bluetooth® LE 具有三种电压检测功能，分别为：POR、BOD 和 LVD。

12.5.3 时钟系统

PSoC™ 4 with AIROC™ Bluetooth® LE 包括下面各时钟源：

- **内部主振荡器 (IMO)**：在 PSoC™ 4 with AIROC™ Bluetooth® LE 中，IMO 是主要的内部时钟源。CPU 和所有高速外设都由 IMO 或外部晶振 (ECO) 提供时钟。PSoC™ 4 with AIROC™ Bluetooth® LE 的多个外设时钟分频器均由 IMO 或 ECO 提供时钟，这些时钟为高速外设提供脉冲。IMO 可以生成频率范围为 3 MHz 到 48 MHz (步长为 1 MHz)，准确度为±2%的时钟。
- **内部慢速振荡器 (ILO)**：ILO 是一个超低功耗的 32 kHz 振荡器，它主要为慢速外设生成时钟，使其 (BLESS 除外) 能在深度睡眠模式下工作 (请参见 **WCO**)。
- **外部晶振 (ECO)**：外部晶振具有一个内置的可调试晶体负载电容，用于生成一个高精度的 24 MHz 时钟。它主要为生成 RF 时钟的 Bluetooth® LE 子系统提供时钟脉冲。高精度 ECO 时钟还可以作为 PSoC™ 4 with AIROC™ Bluetooth® LE 器件高频时钟 (HFCLK) 的时钟源。
- **时钟晶振 (WCO)**：可将 32 kHz WCO 作为一个 LFCLK 源使用 (与 ILO 一起使用)。通过 WCO 可以为 Bluetooth® LE 广播和连接事件保持精确的时间间隔。与 ILO 相同，WCO 在所有模式 (休眠和停止模式除外) 下均可用。

Figure 65 显示的是 PSoC™ 4 with AIROC™ Bluetooth® LE 器件的时钟架构。

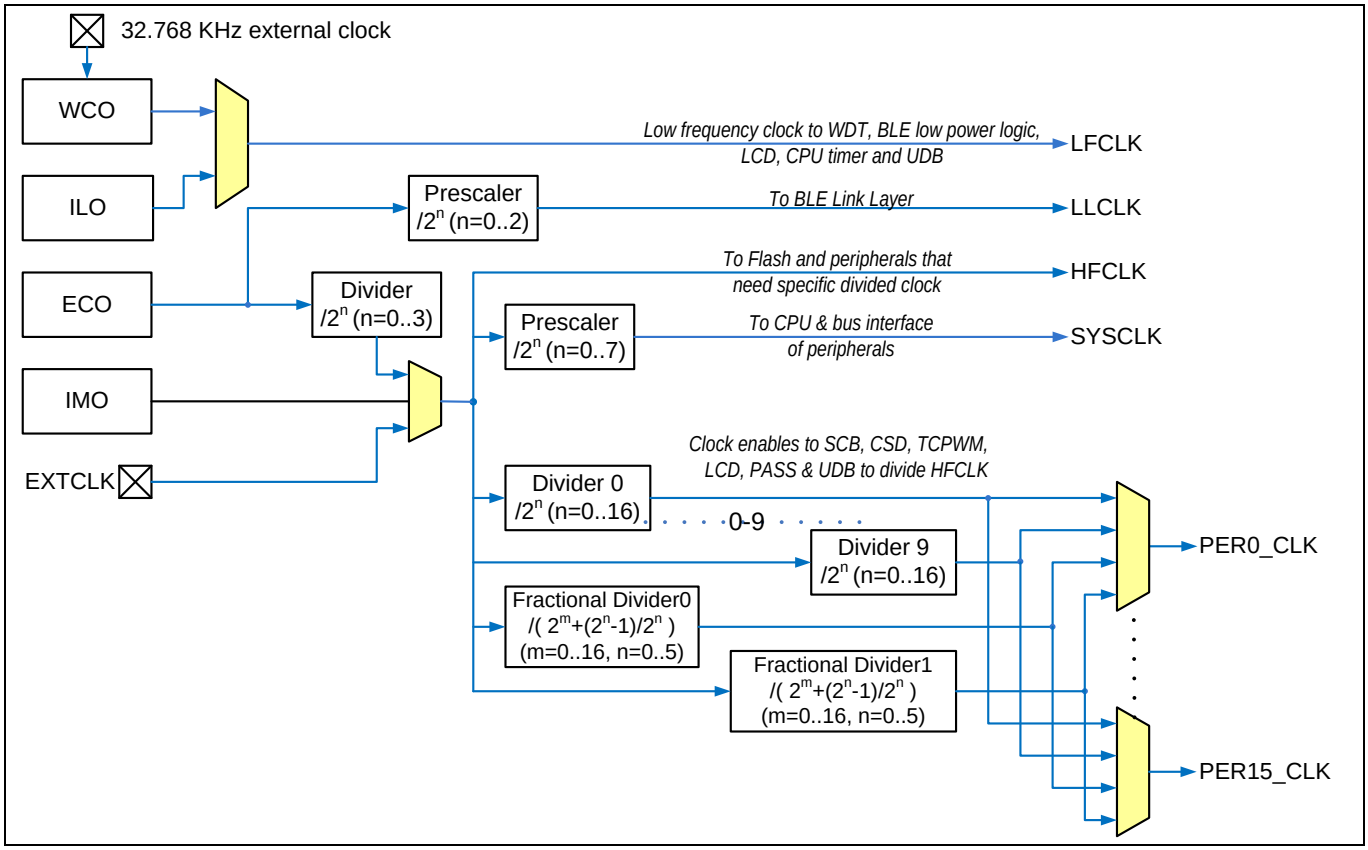


Figure 65 PSoC™ 4 with AIROC™ Bluetooth® LE 时钟系统

12.5.4 器件安全性

PSoC™ 4 with AIROC™ Bluetooth® LE 提供了多个保护闪存存储器的选项，用以避免未经授权的访问或复制。闪存的每一行都有一个保护位；这些位被保存在监管闪存行中。

12.6 可编程 GPIO

I/O 系统提供了一个 CPU 与外设以及外界间的接口。PSoC™ 4 with AIROC™ Bluetooth® LE 具有多达 36 个可编程 GPIO 引脚。您可将 GPIO 配置为 CapSense™、LCD、模拟或数字信号。PSoC™ 4 with AIROC™ Bluetooth® LE GPIO 支持多种驱动模式、驱动强度以及转换速率。

PSoC™ 4 with AIROC™ Bluetooth® LE 提供了一个智能布线系统，从而可以灵活地将内部信号连接至 GPIO。这种灵活的布线简化了电路设计和电路板布局。

13 附录 E：Bluetooth® LE 协议

13.1 概况

Bluetooth® LE，又称 Bluetooth Smart，是由蓝牙 SIG 提供的一个低功耗无线标准，其工作频带为 2.4 GHz ISM。**Figure 66** 显示了 Bluetooth® LE 堆栈。

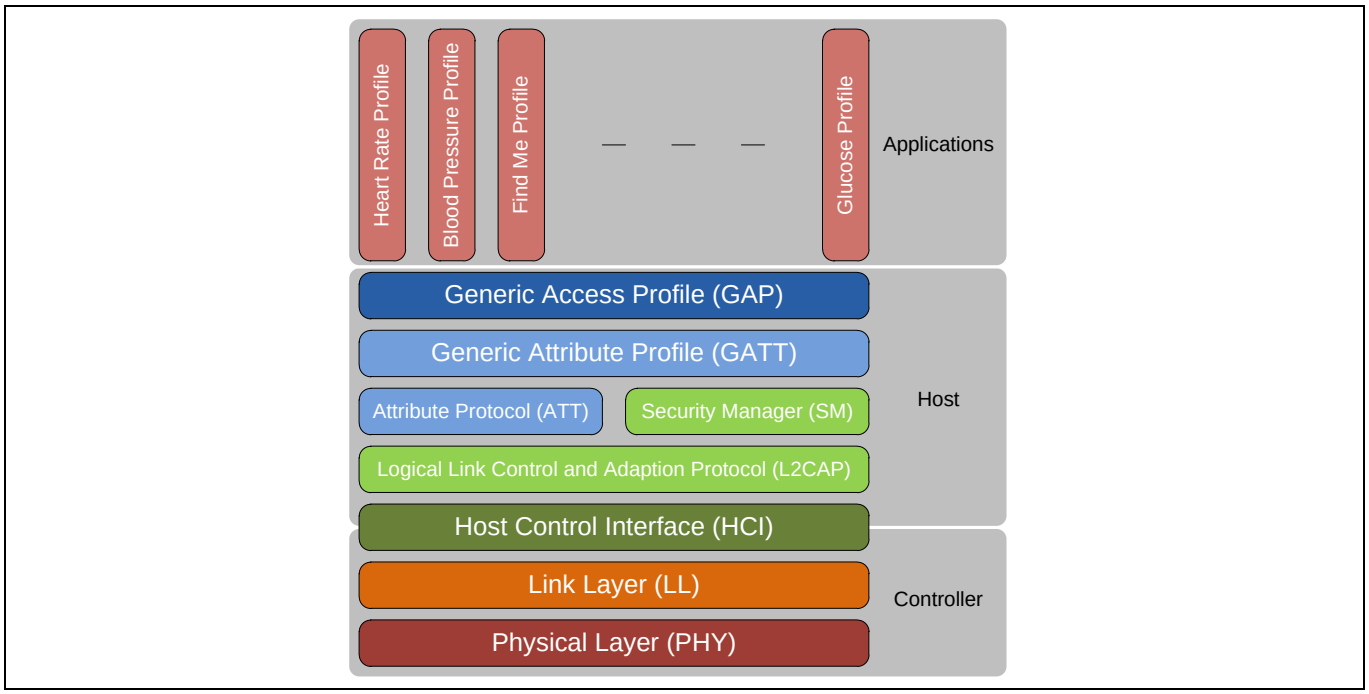


Figure 66 Bluetooth® LE 架构

Bluetooth® LE 堆栈可分为以下三部分：

- **控制器：**它是一个物理器件，用于编码数据包，并把该数据包作为无线信号进行传输。作为接收端使用时，控制器对无线信号进行解码，并重新构建数据包。
- **主机：**软件堆栈包括各种协议和配置文件 (安全管理器、属性协议等)，用于管理两个或多个器件间相互通信的方式。
- **应用：**指的是一个使用场合，其中通过使用软件堆栈和控制器执行特定功能。

下面各节是对 Bluetooth® LE 堆栈中各层的简介，并使用了标准的心率和电池服务作为示例。欲了解 Bluetooth® LE 架构描述的详细信息，请参考蓝牙开发者网页上的蓝牙 4.2 规范或观看相关的培训视频。

13.2 物理层 (PHY)

物理层使用了高斯频移键控 (GFSK) 调制方式在 2.4 GHz ISM 频带上发送并接收数字数据，数据速率为 1 Mbps。Bluetooth® LE 物理层将 ISM 频带分为 40 个 RF 通道 (通道间隔为 2 MHz)，其中有 37 个数据通道，剩下为 3 个广播通道。

13.3 链路层 (LL)

链路层通过执行关键流程来建立一个可靠的物理链路 (通过使用确认和基于流控的架构)，并用于增强 Bluetooth® LE 协议和降低功耗的性能。下面列出的是链路层的几项功能：

- 广播、扫描、创建和保持连接，用以建立物理链路

附录 E：Bluetooth® LE 协议

- 24 位 CRC 和 128 位 AES 加密，用于执行稳定安全的数据交换
- 为低功耗操作创建快速连接和低占空比广播
- 自适应跳频 (AFH)，它改变了用于数据包传输的通信通道，从而降低了其他器件的干扰

在链路层可定义两部分内容：

- **主设备**：以智能手机为例，它能够对主设备配置中的链路层进行配置。
- **从设备**：以心率监视设备为例，它能够对从设备配置中的链路层进行配置。

PSoC/PRoC-BLE 器件可以按照上述任意配置运行。

链路层从设备将它的存在信息广播给另一个链路层主设备。链路层主设备会接收广告数据包，然后根据具体应用的要求选择需要连接的从设备 (请参见 **Figure 67**)。在心率监示器应用的示例中，心率监示器作为从设备使用，它将数据发送给作为主设备的智能手机。这时，智能手机应用将显示智能手机中数据的读取过程。

PSoC/PRoC-BLE 器件在硬件中执行链路层中对时间要求严格并且处理器密集型的任务，如广播、CRC 以及 AES 加密。链路层控制操作 (如进入广播状态和启动加密) 都是通过固件实现的。

Figure 67 显示的是 Bluetooth® LE 链路层数据包的结构以及链路层数据包中各字段的大小。链路层数据包将所有上层数据保存在其负载域内。它具有一个 4 字节的访问地址，可用于识别物理链路上进行的通信，并会忽略正在同一个 RF 通道中运行的相邻 Bluetooth® LE 器件的数据包。24 位 CRC 提供了稳定的数据。

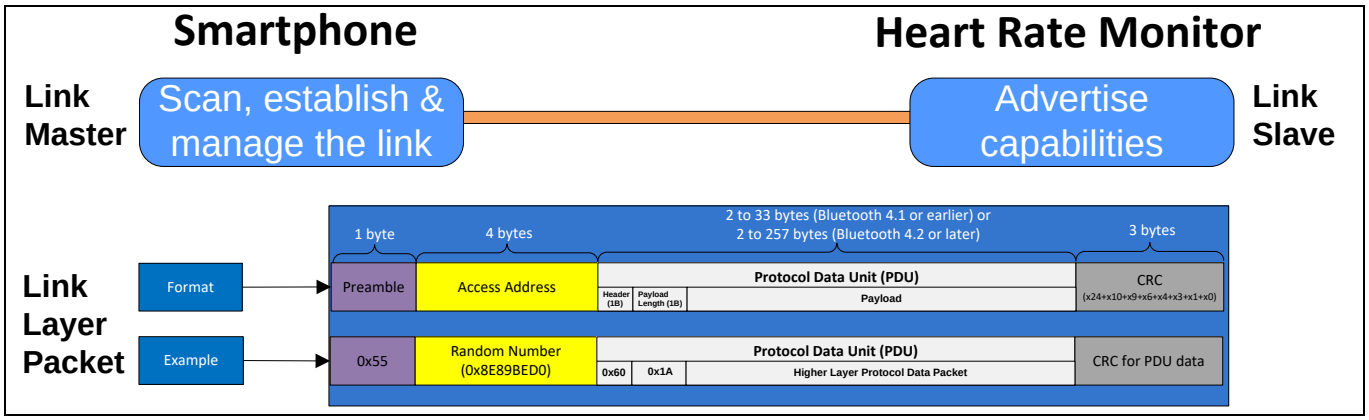


Figure 67 Bluetooth® LE 链路层协议

13.4 主机控制接口 (HCI)

HCI 是主机与控制器间的标准定义接口。通过它，主机和控制器在进行不同的物理传输方式 (如 USB 或 UART) 中可以交换命令、数据和事件等信息。仅在该控制器和主机为不同的器件时，HCI 才需要进行物理传输。

在 PSoC™ 4 with AIROC™ Bluetooth® LE/PRoC-BLE 器件中，HCI 作为一个固件协议层用于控制器和主机间信息和事件的传输。

13.5 逻辑链路控制及适配协议 (L2CAP)

L2CAP 为上层协议提供了协议复用、分段和重组服务。分段功能会将接收到的来自上层的数据包分为链路层可传输的更小的数据包，而重组功能会将接收到的来自链路层的小数据包结合成一个有意义的数据包。L2CAP 层为**属性协议 (ATT)**、**安全管理器 (SM)** 和 L2CAP 控制支持三个协议通道 ID，如 **Figure 68** 所示。蓝牙 4.2 通过这些协议通道顶层上的 L2CAP 定向连接通道允许直的接数据通道。

L2CAP 及其上面各层在 PSoC™ 4 with AIROC™ Bluetooth® LE/PRoC-BLE 器件的固件内实现。

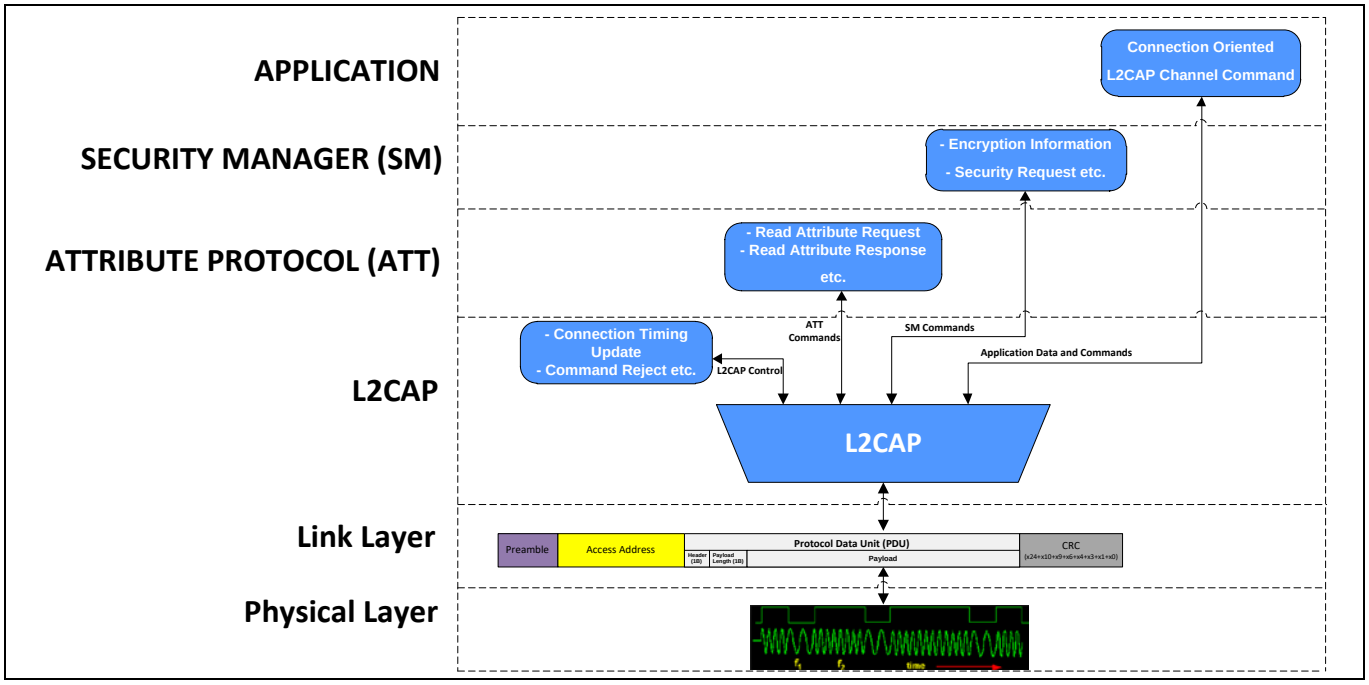


Figure 68 Bluetooth® LE L2CAP 层

13.6 安全管理器 (SM)

SM 层定义了用于配对、加密和关键分配的方法。

- **配对**是使能安全性能的过程。在该过程中，会验证两个器件，对其链接进行加密，然后交换加密密钥。这样可以通过 Bluetooth® LE 接口安全交换数据而不会被 RF 通道上的无声监听器窃听。
- **绑定**是一个过程，其中密钥和识别信息在保存配对过程中被交换。器件绑定完毕后，重新连接这些器件时不需要再次进行配对过程。

Bluetooth® LE 使用了 128 位 **AES** 对数据进行加密。

13.7 属性协议 (ATT)

为了了解 ATT 和 GATT 层，您应该掌握 Bluetooth® LE 中 GATT 的两种使用情况：

- **GATT 服务器：**是指包含数据或信息的设备。它接收 GATT 客户端设备的请求，并使用数据进行响应。例如，心率监示器的 GATT 服务器包含心率信息；Bluetooth® LE HID 键盘的 GATT 服务器包含用户按键信息。
- **GATT 客户端：**是指对 GATT 服务器发送请求和/或接收数据的设备。例如，智能手机是接收心率服务器心率信息的 GATT 客户端；手提电脑是接收 Bluetooth® LE 键盘按键信息的 GATT 客户端。

ATT 构成 Bluetooth® LE 通信基础。该协议使 GATT 客户端能够在 GATT 服务器中找到数据和属性并访问它。有关 GATT 客户端和服务器架构的更多信息，请参考**通用属性配置文件 (GATT)**。

属性包含的是 ATT/GATT 内的基本数据，它包含了以下内容：

- **属性句柄：**这个 16 位的地址用于寻址和访问某项属性。
- **属性类型：**用于指定在某项属性中储存的数据类型。它由一个蓝牙 SIG 定义的 16 位 UUID 表示。

例如，心率服务的 16 位 UUID 表示为 0x180D；器件名称属性的 UUID 表示为 0x2A00。请访问蓝牙网页，以获取 SIG 所分配的一系列 16 位 UUID。

- **属性值：**储存在属性中的实际数据。
- **属性许可：**用于指定属性访问、验证和授权等要求。属性许可由较高层规范设置，不能通过属性协议查看属性许可。

Figure 69 显示的是一个器件名称属性结构示例。

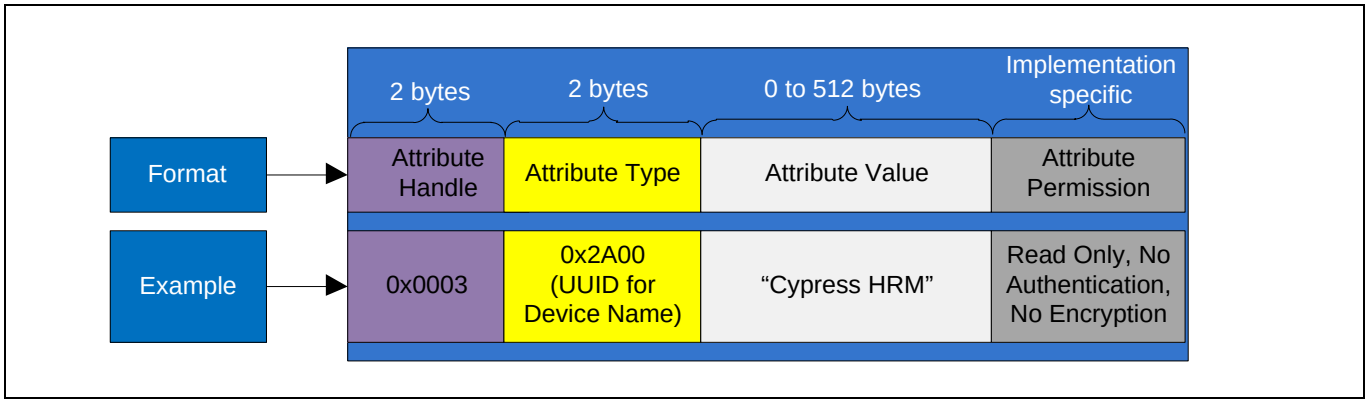


Figure 69　属性格式示例

13.7.1　属性层次结构

属性是用于代表 ATT/GATT 中数据的构建模块。属性大致可分为以下两组，用于提供属性层次结构和数据提取：

- **特性：**显示系统信息或有意义的数据的属性集。一个特性包括以下各属性：
  - 特性声明属性：它定义了一个特性的开始。
  - 特性值属性：它保持了实际数据。
  - 特性描述符属性：它们是可选属性，提供了特性值的其他信息。

“电池电量”是电池服务 (BAS) 中的某个特性示例。以百分比值的形式显示电池电量便是一个特性描述符示例。

Figure 70 显示的是一个特性结构和电池电量特性示例。

  - 某种特性的第一部分是特性声明 (它表示特性的开始) 内容，该声明由 **Figure 70** 中的电池电量指示。
  - 后面是实际的特性值或实际数据，在电池电量特性示例中表示当前电池的电量。电池电量按全量程的百分比形式显示，如‘65’、‘90’等。
  - 特性描述符提供了额外信息，用于描述特性值的意义。例如，电池电量的特性表示格式描述符指示：电池电量被表示为一个百分比值。因此，读取‘90’时，客户端会将其理解为 90%，而不是 90 mV 或 90 mAh。同样，有效范围特性描述符 (不显示在 **Figure 70** 中) 表示电池电量的范围为 0 至 100%。
  - 通常，一个客户端特性配置描述符 (CCCD) 还可作为特性描述符使用，这样通过 GATT 客户端可以配置 GATT 服务器的特性行为。当 GATT 客户端将 0x01 值写入到某个特性的 CCCD 内时，便能够使 GATT 服务器发送异步通知 (将在下一部分进行描述)。在电池电量特性中，将 0x01 值写入到电池电量 CCCD 内会使电池服务定期通知它的电池电量或电池电量值发生的所有变化。

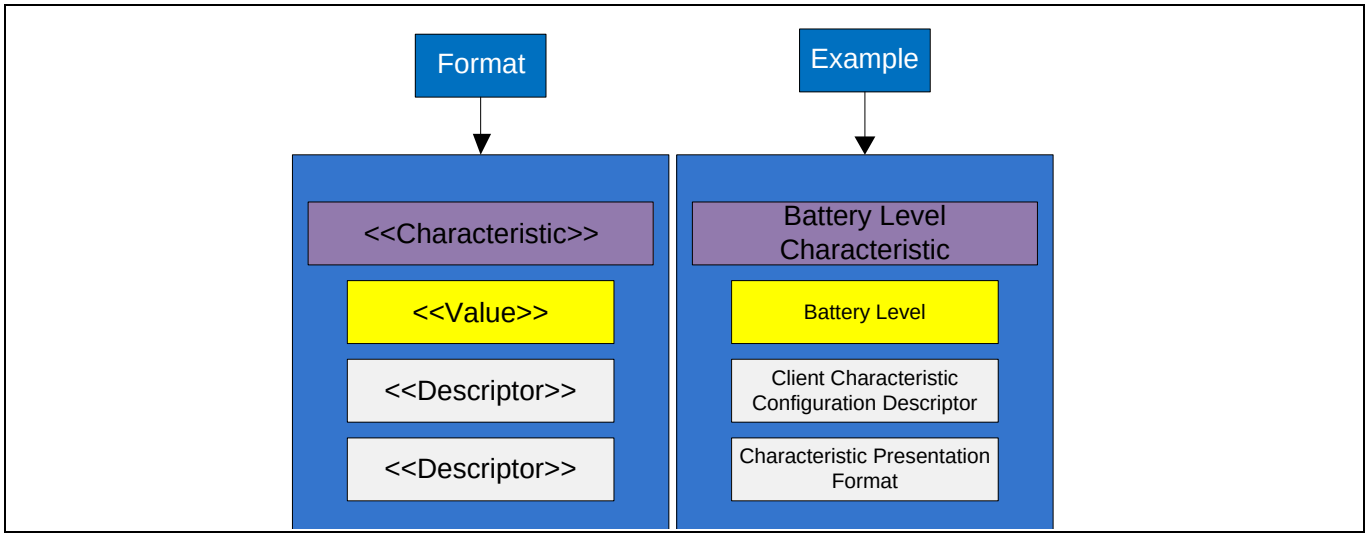


Figure 70　特性格式与示例

- **服务：**属性的类型，它定义了由 GATT 服务器所执行的功能。服务是一个特性集，它也可能包含其他服务。服务用于建立相关数据组并提供数据的层次结构。请参考 **Figure 71**，了解心率服务 (**HRS**) 示例。

服务可分为两种类型：主要服务或辅助服务。主要服务显示了器件的主要功能，而辅助服务则提供了额外功能。例如，在心率监示器中，心率服务 (HRS) 是主要服务，电池服务 (BAS) 是辅助服务。

一个服务还可以包含 GATT 服务器上存在的其他服务。所包含的全部服务组成新服务的一部分。

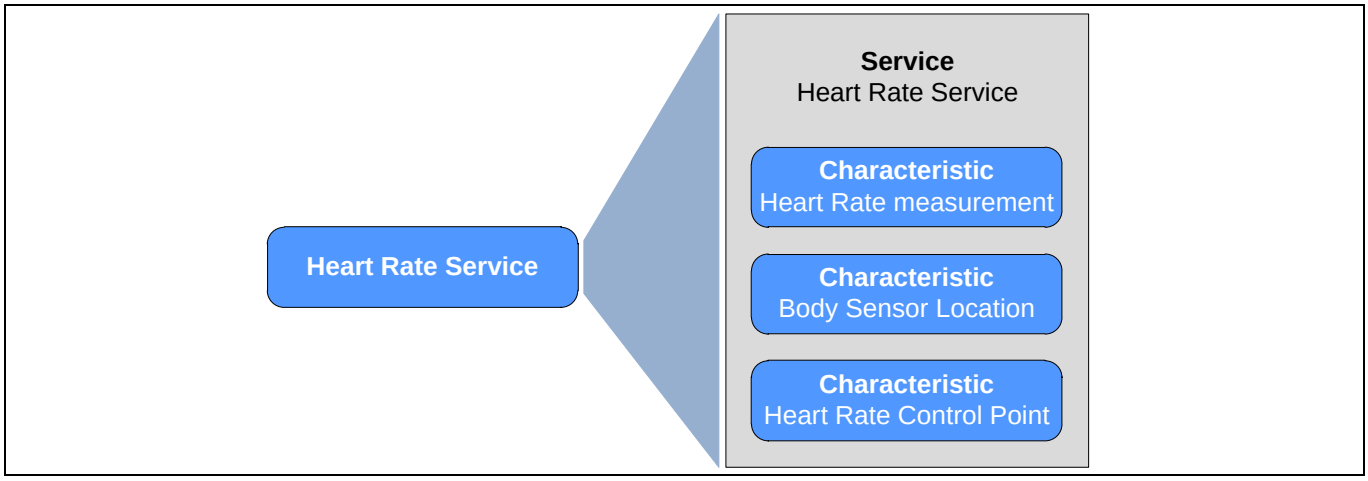


Figure 71　Bluetooth® LE 心率服务示例

Bluetooth® LE 中的“配置文件”指的是一个服务集和这些服务共同执行某个特定终端应用时的行为。心率配置文件 (**HRP**) 是 Bluetooth® LE 配置文件的一个示例，该 Bluetooth® LE 配置文件定义了创建心率监示器所需要的所有服务。更多详细信息，请参考**通用访问配置文件 (GAP)** 章节。

Figure 72 显示的是数据的层次结构，该数据层次结构使用了该章节中所定义的属性、特性、服务和配置文件。

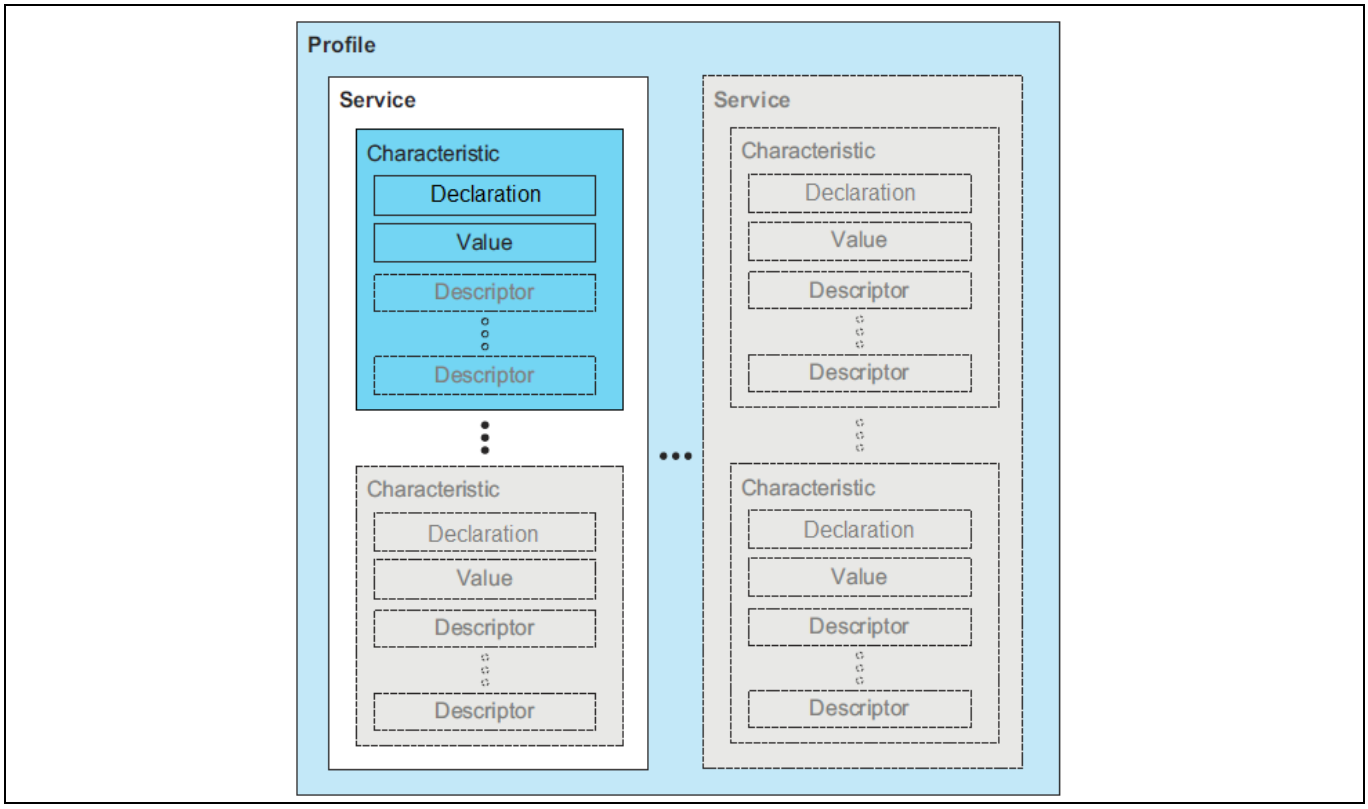


Figure 72　Bluetooth® LE 数据的层次结构*

*图片由蓝牙 SIG 提供

13.7.2　属性操作

通过使用以下五种基本方法，能够访问以上章节所定义的属性：

- **读取请求：**GATT 客户端将该请求发送给 GATT 服务器，以读取属性值。对于所有请求，GATT 服务器会向 GATT 客户端发出一个响应。智能手机读取心率监示器的电池电量 (请参考 **Figure 70**) 便是读取请求的一个示例。
- **写入请求：**GATT 客户端向 GATT 服务器发送该请求，以写入一个属性值。GATT 服务器会向 GATT 客户端发出响应，以表示是否写入了该值。智能手机会将数值 0x01 写入到表示电池电量特性的 CCCD 内，以使能通知作为写入请求的一个示例。
- **写入命令：**GATT 客户端向 GATT 服务器发送该命令以写入一个属性值。对于该命令，GATT 服务器不会发送任何响应。例如，Bluetooth® LE 即时警报服务 (**IAS**) 使用一个写命令从一个 IAS 定位器 (例如，一部智能手机) 来触发 IAS 目标器件 (例如，使用 Bluetooth® LE 的密钥卡) 上的警报 (打开 LED、使蜂鸣器报鸣、驱动振动电机等) 。
- **通知：**GATT 服务器将其发送给 GATT 客户端，从而通知属性新值。GATT 客户端不会对一个通知发送任何响应。例如，将数值 0x01 写入到心率监示器的 CCCD 内时，该心率监示器会向智能手机发送心率测量通知。
- **指示：**GATT 服务器负责发送该类型的信息。GATT 客户端始终会确认该信息。例如，Bluetooth® LE 健康温度计服务 (**HTS**) 使用各指示可靠地向健康温度计收集器 (如智能手机) 发送所测量的温度值。

13.8 通用属性配置文件 (GATT)

GATT 定义了查找和使用属性的方式。GATT 可以按下面两种方式中的一种进行操作：

- **GATT 客户端：**是指请求数据的器件 (例如智能手机)
- **GATT 服务器：**是指提供数据的器件 (例如心率监示器)

Figure 73 显示了 GATT 层上的客户端-服务器架构，它以心率监测设备为例。该心率监示器提供了多项服务 (HRS、BAS 和器件信息服务) 。每项服务都包含一个或多个带有特性值和描述符的特性，如 **Figure 70** 所示。

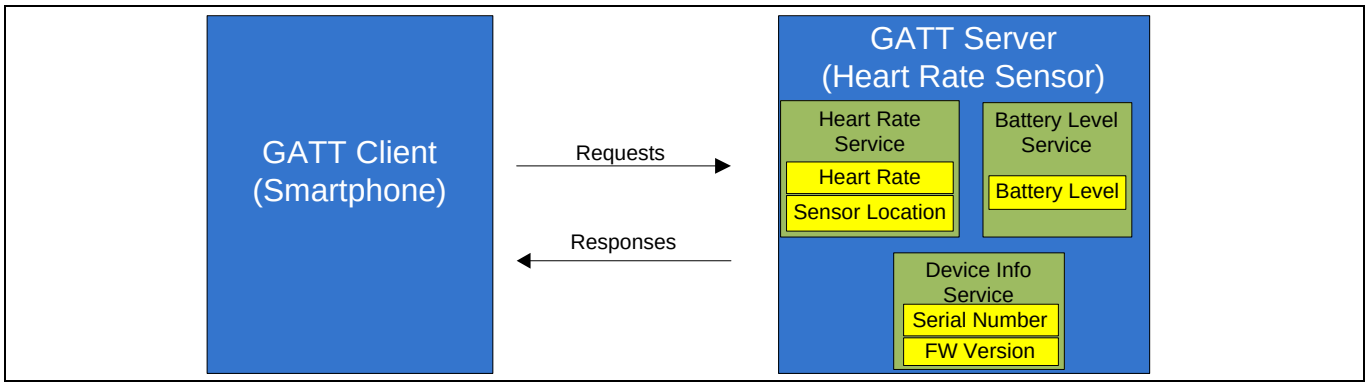


Figure 73 GATT 客户端-服务器架构

以链路层级别建立 Bluetooth® LE 连接后，GATT 客户端 (最初并不了解已连接的 Bluetooth® LE 器件) 会启动一个被称为“服务发现”的过程。作为服务发现过程的一部分，GATT 客户端会向 GATT 服务器发送多个请求，从而获取 GATT 服务器中包含了所有可用服务、特性以及属性的列表。服务发现过程完成后，GATT 客户端将具有所需信息，从而通过使用前一章节所介绍的属性操作来修改或读取 GATT 服务器所显示的信息。

13.9 通用访问配置文件 (GAP)

GAP 层提供了设备特定的信息，如设备地址、设备名称以及发现、连接、绑定的各种方法。配置文件定义了一个设备的发现和连接方法、可用服务的列表，以及服务的使用方式。**Figure 75** 显示的是一个心率配置文件示例。

GAP 可以按照下面四种方式中的一种进行操作：

- **外设：**它起着广播作用，使设备能够连接至 GAP 中心。建立与 GAP 中心的连接后，器件会作为从设备进行操作。例如，心率传感器会将所测量的心率报告给远程设备，使之作为一个 GAP 外设进行操作。
- **中心：**它作为 GAP 使用，用于扫描广播和启动同外设间的连接。建立与外设的连接后，该 GAP 作为主设备运行。例如，一个外设 (心率传感器) 检测心率测量数据的智能手机作为 GAP 中心操作。
- **广播员：**它作为一个发送器，用于发送数据该功能不能实现 Bluetooth® LE 连接和进行数据交换 (无请求/响应操作) 它的工作原理类似于一个广播电台：它会不断发送数据，却不需关心是否有人收听，它进行的是单向数据通信。作为一个信标是 GAP 广播员的典型示例，它会不断播放信息而无需任何响应。
- **观察员：**作为一个“听众”，它会扫描广播，但不会连接到广播的设备。观察员角色与广播员角色完全相反。它的工作原理类似于一个无线收音机，可以一直收听信息而不会向信息源发送任何响应。智能手机中可连续监听各种信标的应用便是一个 GAP 观察器的典型示例。

Figure 74 显示的是一个通用 Bluetooth® LE 系统，其中，Infineon 的 Bluetooth® LE Pioneer 套件作为外设，一台智能手机作为中心设备。此外，该图还说明了各 Bluetooth® LE 协议层间的交互以及它们在中心设备和外设中所起的作用。

附录 E：Bluetooth® LE 协议

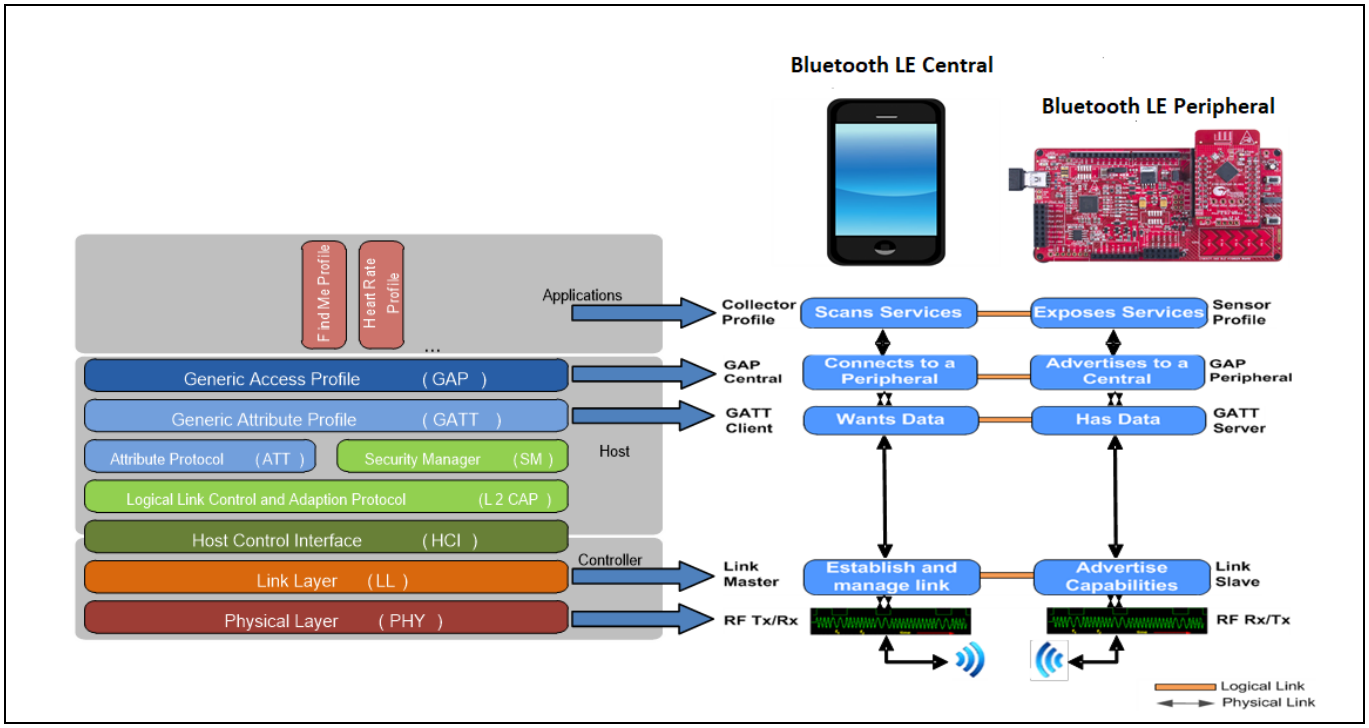


Figure 74 Bluetooth® LE 系统设计

Figure 75 显示的是一个示例，其中，一个带有心率应用的智能手机作为中心设备使用，另外一个心率传感器作为外设使用。监示心率的设备将执行心率传感器的配置文件，而接收数据的智能手机会执行心率收集器的配置文件。

在该示例中，心率传感器配置文件实现了两项标准的服务。第一项是心率服务，它包含三项功能，分别为心率测量功能、身体感应位置功能和心率控制点功能。第二项是设备信息服务。在链路层上，心率测量器件作为从设备，智能手机作为主设备。有关心率服务和配置文件的详细消息，请参见蓝牙开发人员的入口网站。

附录 E：Bluetooth® LE 协议

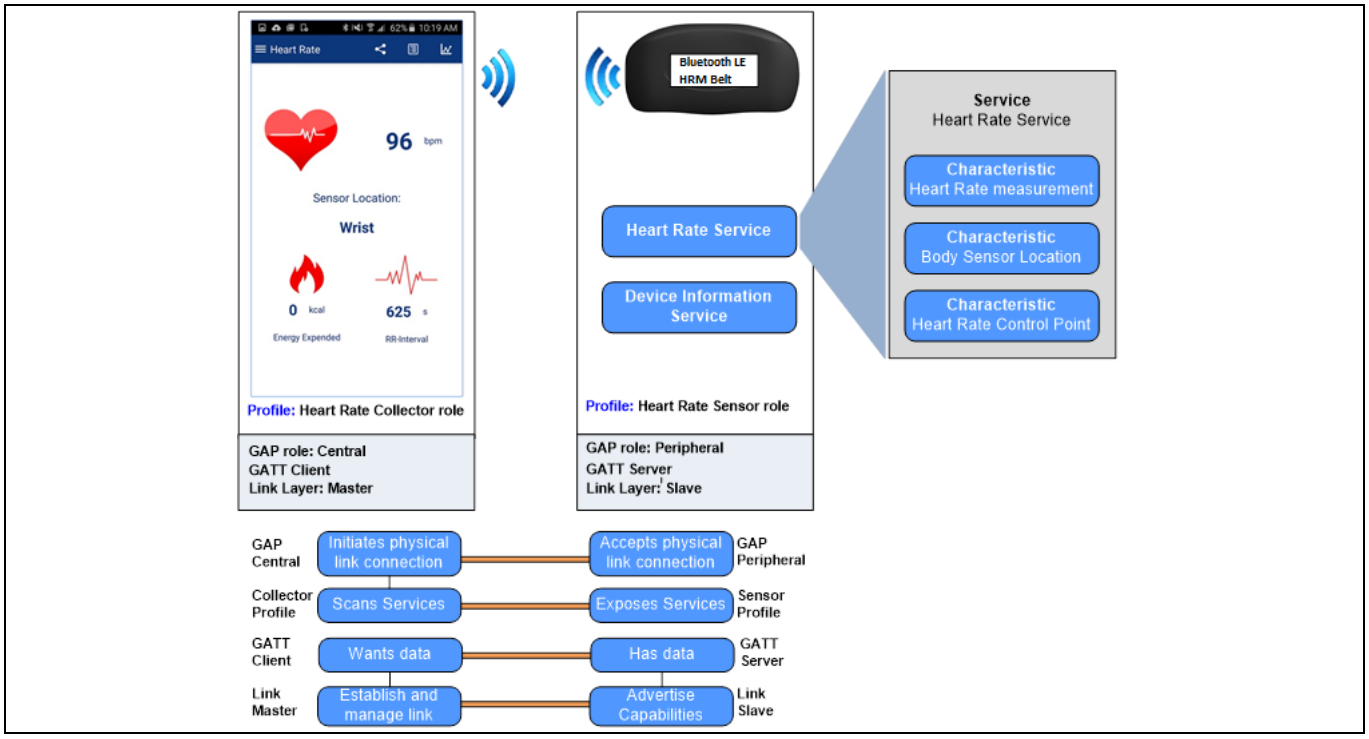


Figure 75 Bluetooth® LE 心率监测系统

相关应用笔记

[1] **AN94020 — PRoC Bluetooth® Low Energy 入门**

[2] **AN96841 - Getting started with EZ-Bluetotth® LE Creator modules**

[3] **AN91184 — PSoC™ 4 BLE：设计 Bluetooth® Low Energy 应用**

[4] **AN91162 — 创建自定义的 Bluetooth® Low Energy 配置文件**

[5] **AN92584 — 低功耗设计和估计 Bluetooth® Low Energy 应用的电池使用寿命**

[6] **AN95089 — PSoC™ 4/PRoC Bluetooth® Low Energy 晶体振荡器选择和调校技术**

[7] **AN91445 - 天线设计和射频布局指南**

[8] **AN97060 — PSoC™ 4 BLE 和 PRoC Bluetooth® LE：空中传送 (OTA) 进行设备固件更新 (DFU) 指南**

文档修订记录

Document version	Date of release	Description of changes
**	2015-03-06	本文档版本号为 Rev**，译自英文版 001-91267 Rev*A。
*A	2015-07-06	本文档版本号为 Rev*A，译自英文版 001-91267 Rev*C。
*B	2015-12-16	本文档版本号为 Rev*B，译自英文版 001-91267 Rev*F。
*C	2017-04-21	更新 logo 和版权信息。
*D	2018-02-21	本文档版本号为 Rev. *D，译自英文版 001-91267 Rev. *H。
*E	2021-07-07	翻译自: 001-91267 Rev. *I

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

Edition 2021-07-07
Published by
Infineon Technologies AG
81726 Munich, Germany

© 2021 Infineon Technologies AG. All Rights Reserved.

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

Document reference
001-96121 Rev. *E

重要提示

本文档所提供的任何信息**绝不应当**被视为针对任何条件或者品质而做出的保证（质量保证）。英飞凌对于本文档中所提及的任何事例、提示或者任何特定数值及/或任何关于产品应用方面的信息均在此明确声明其不承担任何保证或者责任，包括但不限于其不侵犯任何第三方知识产权的保证均在此排除。

此外，本文档所提供的任何信息均取决于客户履行本文档所载明的义务和客户遵守适用于客户产品以及与客户对于英飞凌产品的应用所相关的任何法律要求、规范和标准。
本文档所含的数据仅供经过专业技术培训的人员使用。客户自身的技术部门有义务对于产品是否适宜于其预期的应用和针对该等应用而言本文档中所提供的信息是否充分自行予以评估。

如需产品、技术、交付条款和条件以及价格等进一步信息，请向离您最近的英飞凌科技办公室接洽(**www.infineon.com**)。

警告事项

由于技术所需产品可能含有危险物质。如需了解该等物质的类型，请向离您最近的英飞凌科技办公室接洽。

除非由经英飞凌科技授权代表签署的书面文件中做出另行明确批准的情况外，英飞凌科技的产品不应当被用于任何一项一旦产品失效或者产品使用的后果可被合理地预料到可能导致人身伤害的任何应用领域。