请注意赛普拉斯已正式并入英飞凌科技公司。
此封面页之后的文件标注有“赛普拉斯”的文件即该产品为此公司最初开发的。请注意作为英飞凌产品组合的部分，英飞凌将继续为新的及现有客户提供该产品。

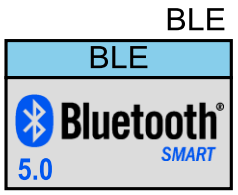
文件内容的连续性
事实是英飞凌提供如下产品作为英飞凌产品组合的部分不会带来对于此文件的任何变更。未来的变更将在恰当的时候发生，且任何变更将在历史页面记录。

订购零件编号的连续性
英飞凌继续支持现有零件编号的使用。下单时请继续使用数据表中的订购零件编号。

低功耗蓝牙（BLE_PDL）

2.0

特性



- 在任意模式组合中，多链接特性最多支持同时进行四个连接
- 支持符合蓝牙版本 5.0 的协议栈
- 通用访问配置文件（GAP）和通用属性配置文件（GATT）
- 安全管理器
- 逻辑链接自适应协议（L2CAP）面向连接通道
- 链路层（LL）

概述

低功耗蓝牙（BLE）外设驱动库（PDL）组件提供了一个基于 GUI 的全面配置窗口，这样便于设计需要 BLE 连接的应用。BLE_PDL 组件包含符合蓝牙内核规范版本 5.0 的协议栈，并提供了各种 API 函数，从而允许用户应用通过协议栈访问基础硬件。

BLE_PDL 组件结合了图形配置实体和一组构建在 PDL 中的 BLE 中间件上的 API。它允许组件配置对话框所定义的基于原理图的各连接和硬件配置。

使用 BLE_PDL 组件的时间

BLE 用于超低功耗网络和物联网（IoT）解决方案，旨在降低电池供电设备的成本，使其能够快速连接并构成简单的无线链接。BLE 组件还支持 HID、遥控、运动和健身监视器、便携式医疗设备以及智能手机配件等各种目标应用。

SIG 所采用的配置文件和服务

BLE_PDL 组件支持许多 SIG 采用的基于 GATT 的配置文件和服务，并可以将其任意匹配到 GATT 客户端或 GATT 服务器。该组件生成特殊配置文件/服务操作需要的所有代码，如 Component Configure 对话框中所配置的内容。

完整的 API

BLE_PDL 组件和 BLE 中间件库提供了应用级 API，以供设计使用，而无需进行手动的协议栈级配置。BLE 中间件库 API 文档会在单独的 HTML 文件中提供。

自定义配置文件

您可以创建使用现有服务的自定义配置文件，也可以创建带有自定义特性和描述符的自定义服务。

调试支持

为了进行测试与调试，可以通过组件嵌入式 UART 将该组件配置为 HCI 模式。请参见 General 选项卡 — BLE Controller only (HCI over UART)项。

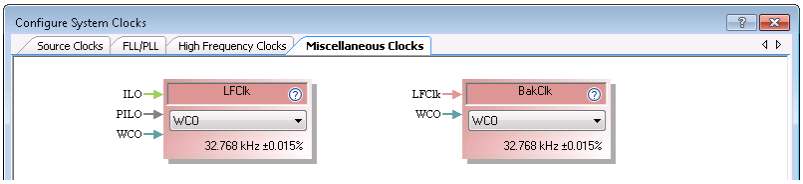
对于无线验证操作，可以将赛普拉斯 CySmart 中心仿真工具用于通用蓝牙主机协议栈仿真。为了启动该工具，请右键单击 Component，然后选择 **Launch CySmart** 项。

快速入门

1. 从组件目录中的 Cypress/Communications 文件夹内，将 BLE_PDL 组件拖到原理图上。
2. 在工作区浏览器中，双击设计范围资源（*<project>.cydwr*）文件树形结构下的 **Clocks** 项以打开时钟编辑器。然后双击表中的 WCO 时钟，以打开 Configure System Clocks 对话框。
   a. 在 **Source Clocks** 选项卡下，启用 **WCO** 时钟。



b. 然后在 **Miscellaneous Clocks** 选项卡下，将“WCO”设置为低频（LFClk）时钟和备份（BakClk）时钟的时钟源。



c. 关闭 Configure System Clocks 对话框。

3. 在*<project>.cydwr* 文件中，选择 **Interrupts** 选项卡。

a. 将 BLE 中断（BLE_bless_isr）分配给运行 BLE 控制器相应的 ARM 内核。它取决于在 **CPU Core** 字段中的选项（请参见 General 选项卡）。

- 对于 **Single core (Complete Component on CM0+)**项，分配给 CM0+
- 对于 **Single core (Complete Component on CM4)**项，分配给 CM4
- 对于 **Dual core (Controller on CM0+, Host and Profiles on CM4)** 项，BLE_bless_interrupt 分配给 CM0+，其他外设中断分配给 CM4

b. 将 **Priority** 设置为最高值。

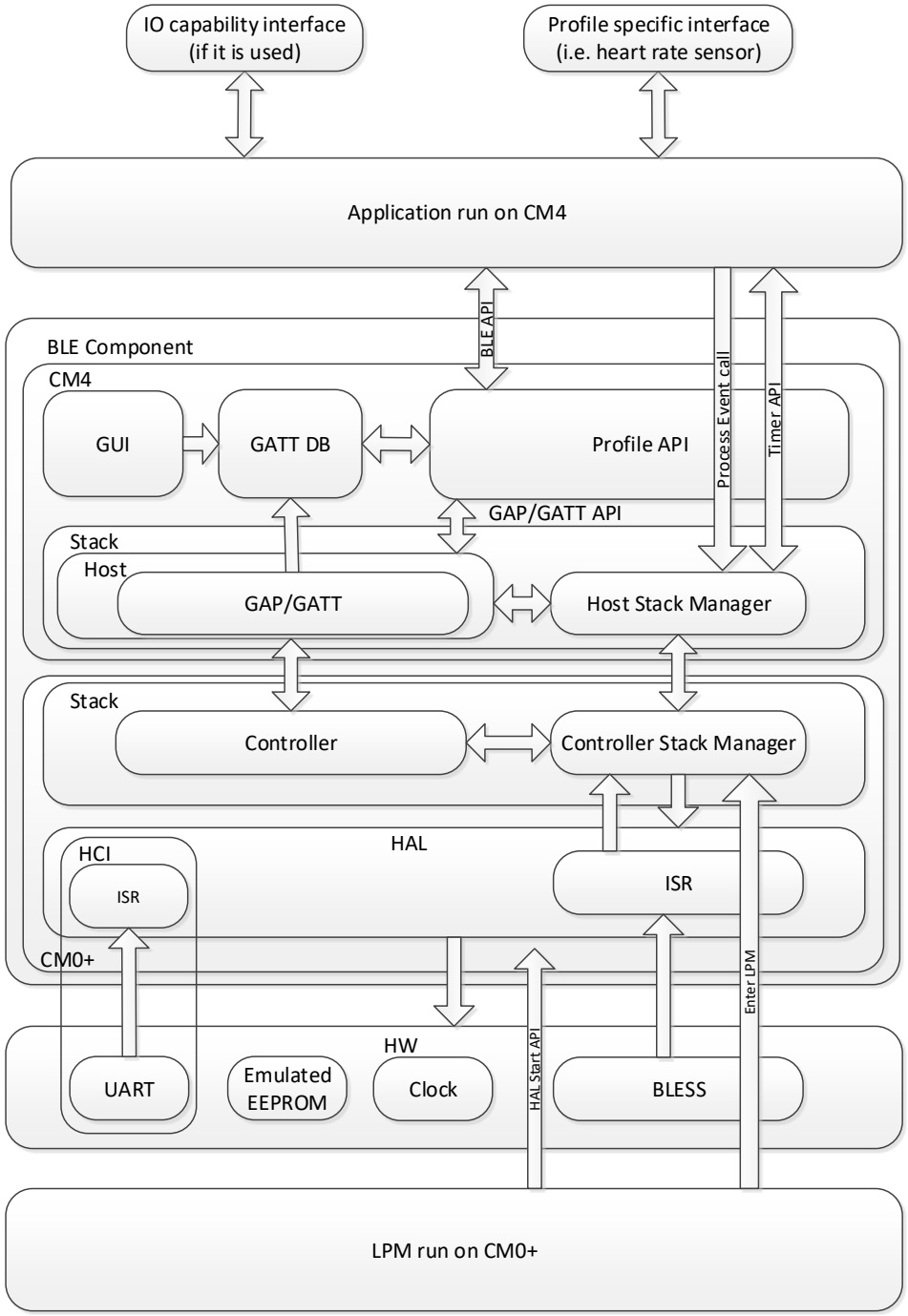
4. 单击 **Build** 以生成 API。一旦时钟和中断配置满足要求，便不会出现警告或错误信息。

注意：如果在 BLE Configure 对话框中将 **CPU Core** 参数设置为 Dual core(Controller on CM0+, Host and Profiles on CM4)项，则必须修改 IPC 配置文件（*cy_ipc_config.h*）（位于工作区的 Shared Files 文件夹中）。应该将定义 CY_IPC_INTR_CYPIPE_MUX_EP0 更改为与深度睡眠唤醒相兼容的中断向量。有效范围为：0..7。

注意： BLE_PDL 组件使用 cy_em_eeprom 来存储绑定数据。为了匹配使用 Em_EEPROM 组件/中间件与 BLE_PDL，请修改链接器脚本。更多信息，请参阅 PDL 文档的中间件/赛普拉斯 Em_EEPROM 中间件部分。要访问该文档，请转至 PSoC Creator **Help menu > Documentation > Peripheral Driver Library**。

BLE_PDL 组件架构

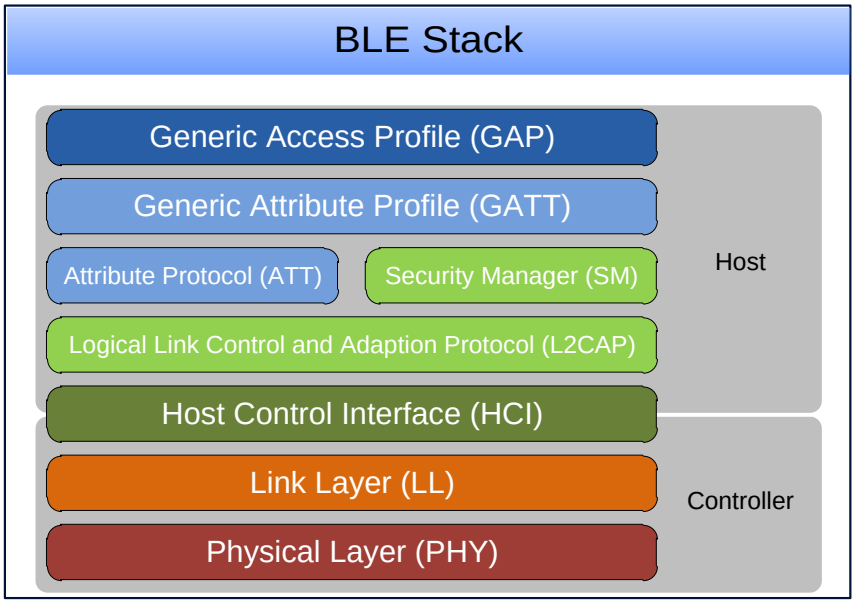
BLE_PDL 组件包括 BLE 协议栈、BLE 配置文件、BLE_PDL 组件硬件抽象层（HAL）以及链路层。下图显示的是 BLE_PDL 组件的上层架构，用于说明各层间的关系以及应用与组件交互的路由。请注意，通过使用回调函数可以向应用通知 BLE 事件。您可以使用这些内容来构建状态机。更多详细信息，请参考回调函数部分。



下面几部分对各层进行了介绍：

BLE 协议栈

BLE 协议栈实现了蓝牙内核规范版本 5.0 中所定义的内核 BLE 功能。该协议栈包含在 BLE 中间件库中，作为预编译库使用。BLE 协议栈能够实现 BLE 协议协议栈的多层架构，如下图所示。



通用访问配置文件（GAP）

通用访问配置文件定义了一些通用程序，包括蓝牙器件的发现以及连接至各个蓝牙器件的链路管理等内容。另外，该配置文件中还包含了对某些参数的普通格式要求（可以使用用户界面访问这些参数）。

通过 LE 物理通道进行操作时，通用访问配置文件会定义以下功能：

- **发送器功能：**作为发送器件运行，用于发送广播事件。将其称为广播器。它有一个发送器，并且可能有一个接收器。
- **观察器功能：**作为观察器运行，用于接收广播事件。将其作为观察器。它有一个接收器，并且可能有一个发送器。
- **外设功能：**通过使用连接建立程序来接收 LE 物理链接的器件被称为“外设功能”。作为外设功能运行的器件将作为链路层连接状态的“从设备功能”。作为外设功能运行的器件被称为外设。外设具有一个发送器和一个接收器。
- **中心器件功能：**支持中心处理功能的器件可初始化物理连接的建立。作为“中心器件功能”运行的器件将作为链路层连接中的“主设备功能”。以中心器件功能运行的器件被称为中心器件。中心器件具有一个发送器和一个接收器。

通用属性配置文件（GATT）

通用属性配置文件使用 ATT 协议层对普通服务框架进行定义。该框架定义了这些服务的规程和格式，以及它们各自的特性。它定义了服务、特性、描述符检测和读取、编写、通知、指示特性以及配置特性广播的规程。

GATT 模式

- GATT 客户端：该器件接收数据。它负责初始化面向 GATT 服务器的指令和请求。它还可以接收 GATT 服务器发送的响应、指示和通知数据。
- GATT 服务器：该器件发送数据，它接收自 GATT 客户端传入的指令和请求，并将响应、指示和通知发送给 GATT 客户端。

BLE 协议栈可以同时支持这两种模式。

属性协议（ATT）

属性协议层定义了一个使用 BLE 逻辑传输通道的客户端/服务器架构。属性协议允许 GATT 服务器将属性集以及它们的相关值开放给 GATT 客户端。GATT 服务器公开的这些属性是由 GATT 客户端检测、读取和写入的，并由 GATT 服务器指示和发送通知。对这些属性进行的传输是一次性的。

安全管理器协议（SMP）

安全管理器协议定义了各种规程和性能，用以管理各个器件间的配对、认证和加密过程。具体包括：

- 加密与认证
- 配对与绑定
  - Pass Key 与 Out of band 绑定
- 用于器件识别分辨率、数据信号和加密的密钥生成
- 根据 GAP 中心和 GAP 外设器件的 IO 功能选择配对方法。

逻辑链路控制适配协议（L2CAP）

L2CAP 提供了一个无连接的数据通道。LE L2CAP 提供了以下各项功能：

- 通道复用 — 它管理三个固定的通道。其中两个通道专用于更高的协议层，如 ATT 和 SMP。另一个通道用于 LE-L2CAP 协议信号通道。
- 它支持数据包的分段和重组，这些数据包的大小等于 BLE 控制器所管理的最大数据包大小。

- 它通过一个特殊应用支持面向连接的通道，该应用使用 PSM（协议服务复用器）通道进行注册。它实现了在两个 LE L2CAP 实体之间根据要求进行流量控制。该功能适用于需要传输大数据量的 BLE 应用。

主机控制器接口（HCI）

HCI 层通过实现一个指令、事件和数据的接口来允许从上层（如 GAP、L2CAP 和 SMP）对链路层进行访问。

链路层（LL）

LL 协议管理着各器件间的物理 BLE 连接。它支持所有 LL 状态，如广播、扫描、初始化和连接（主设备和从设备间）。它实现了所有关键链接控制规程，如 LE 加密、LE 连接更新、LE 通道更新和 LE Ping。链路层是由硬件和固件联合实现的，其中高速 LL 的关键功能是在 LL 硬件中实现的。LL 固件保持并控制着关键的 LL 规程状态机。它支持所有特定于 BLE 芯片的低功耗模式。

BLE 协议栈是 BLE_PDL 组件解决方案中的预编译库。在编译过程中，会根据应用链接 BLE 协议栈库的合适配置。BLE 协议栈库符合 ARM 嵌入式应用二进制接口（EABI）的标准，通过使用 ARM 编译器版本 5.03 可以对这些库进行编译。

下表显示的是 BLE 协议栈库与用户配置的完整 BLE 协议模式或 HCI 模式的映射状况。有关协议栈配置的选择，请参考 General 选项卡部分。

BLE_PDL组件配置	CPU内核	BLE协议栈库
完整的BLE协议 具有软件接口的单核上的主机和控制器	单核（CM4上的完整组件）	cy_ble_stack_gcc_radio_max_cm4.a cy_ble_stack_gcc_soc_cm4.a
	单核（CM0+上的完整组件）	cy_ble_stack_gcc_radio_max_cm0p.a cy_ble_stack_gcc_soc_cm0p.a
完整的BLE协议 具有IPC接口的双核上的主机和控制器	双核（CM0 +上的控制器和CM4上的主机及配置文件）	cy_ble_stack_gcc_radio_max_cm0p.a cy_ble_stack_gcc_controller_ipc_cm0p.a cy_ble_stack_gcc_host_ipc_cm4.a
HCI模式	CM4	cy_ble_stack_gcc_radio_max_cm4.a cy_ble_stack_gcc_controller_uart_cm4.a
	CM0+	cy_ble_stack_gcc_radio_max_cm0p.a cy_ble_stack_gcc_controller_uart_cm0p.a

可以在 **General** 选项卡中配置协议栈模式和内核参数。要想成功构建，请确保在设计范围资源的中断编辑器（*project.cydwr*）中将 BLE_bless_isr 分配给控制器所在的内核。

有两组库：

- “_gcc_” libraries are built with wchar_t typedef 设置为 32 位。
- “_mdk_” libraries are built with wchar_t 设置为 16 位。

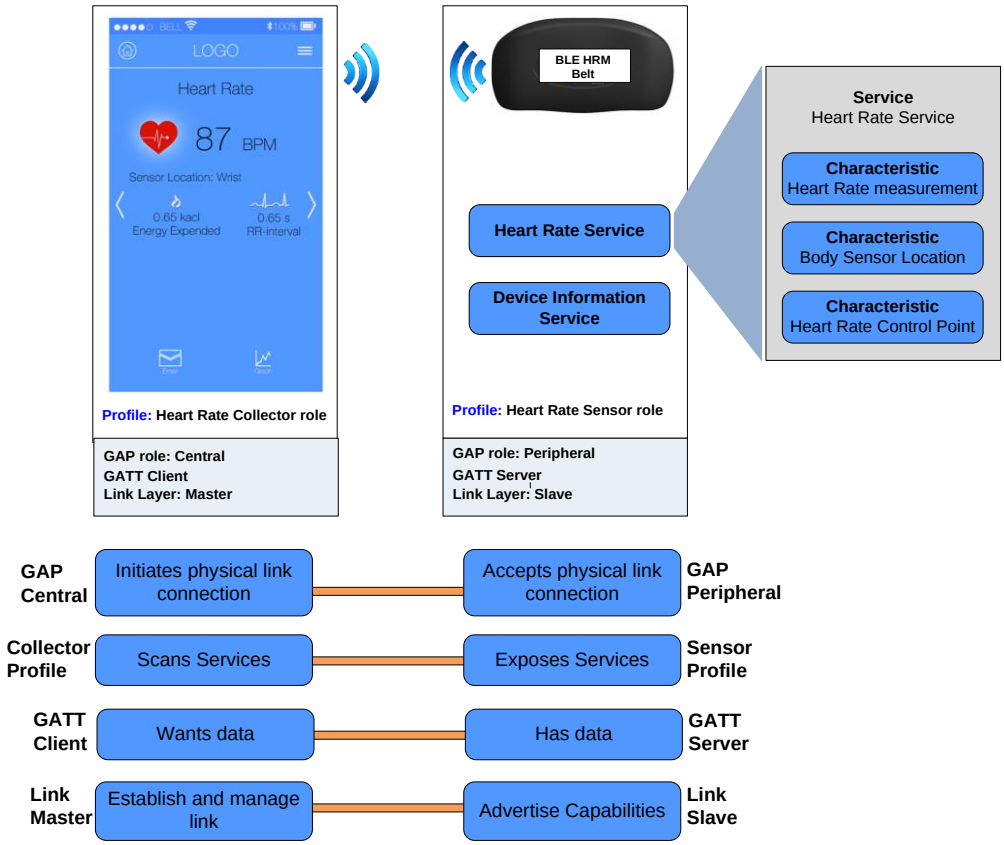
注意： “_gcc_”库用于 IAR 编译器。

配置文件层

在 BLE 中，数据被划分为配置文件、服务和特性等类型。

- **Profile**（配置文件）介绍的是设备间的连接方法，用以查找并使用‘Service’（服务）。蓝牙设备使用该定义描述了应用类型以及设备的常规预期行为。有关如何配置 BLE_PDL 组件的信息，请参阅组件参数部分。
- **服务**是指被称为‘Characteristic’（特性）的数据对象集。通过服务来定义配置文件中的具体函数。也可以使用某个服务来定义它与其它服务间的关系。将通用惟一标识符（UUID）分配给某个服务。它是一个 16 位（针对 SIG 采用的服务）或 128 位（针对自定义服务）的标识符。更多有关将服务添加到配置文件的信息，请参考工具栏部分。
- **Characteristic**（特性）包含一个数值和描述符，该描述符用于描述特性值。它是某项服务中特定信息的一种属性类型。与某个服务相同，每种特性都被指定了一个 UUID（针对 SIG 所采用的特性和自定义特性分别是 16 位和 128 位）。更多有关特性和描述符的详细信息，请参考工具栏部分。

下图显示的是简单的 BLE 心率监测仪应用中的配置文件、服务和特性间的关系（该应用使用了心率配置文件）。



心率配置文件包含一个心率服务和一个器件信息服务。心率服务包含 3 项特性，每项特性包含不同的信息。将图中的器件配置为一个传感器，即在心率配置文件的上下文中，器件作为一个 GAP 外设和 GATT 服务器。在 BLE 协议栈说明中介绍了这些概念。

PSoC Creator 通过使用指定在 GUI 中的参数配置生成配置文件层。配置文件能够为应用实现所需要的配置文件特定属性数据库和 API。您可以选择配置标准 SIG 所采用的配置文件并生成一个设计或定义应用所需要的自定义配置文件。GUI 允许将配置文件设计以 XML 格式导入/导出，以重新使用。此外，还提供了符合 Bluetooth Developer Studio 的 XML 格式文件。

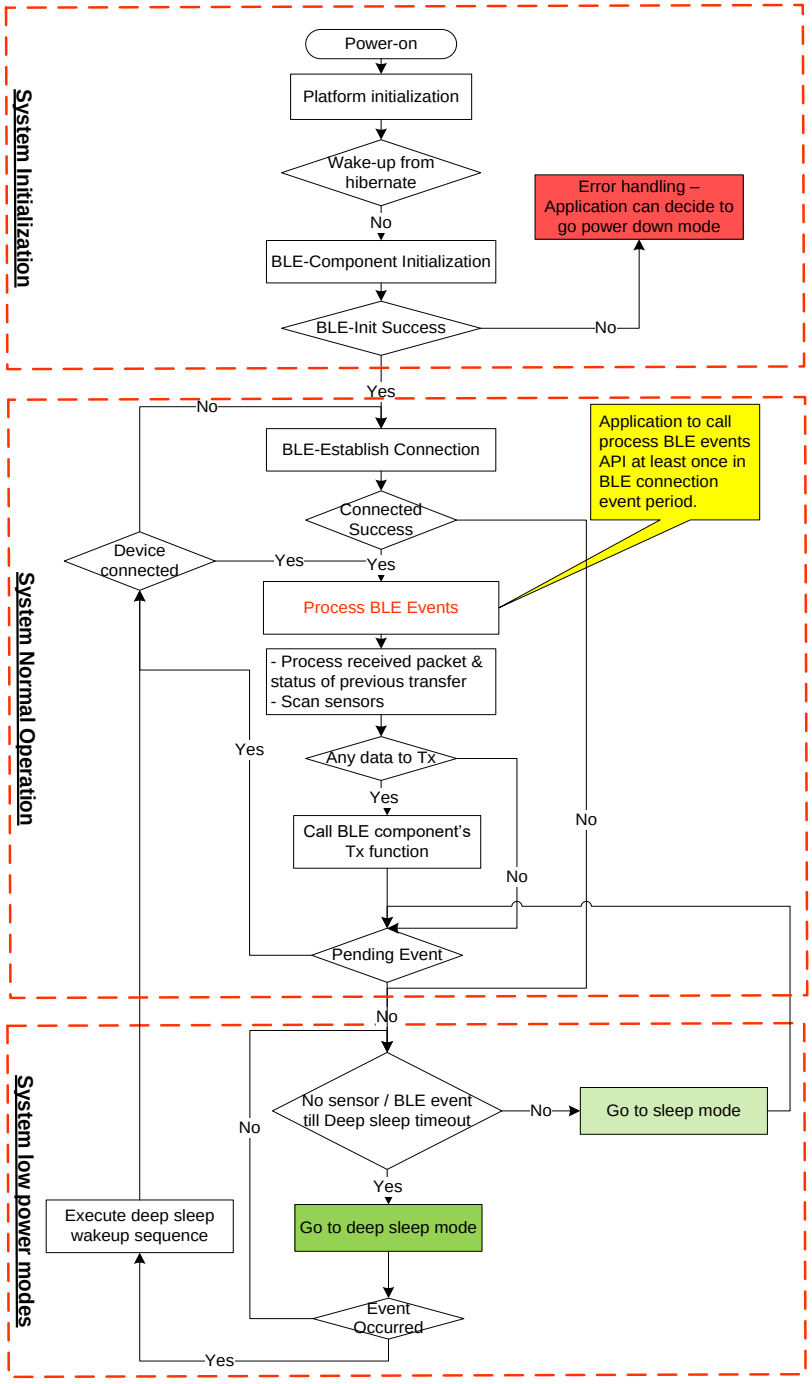
硬件抽象层（HAL）

HAL 实现 BLE 协议栈和基础硬件之间的连接。该层仅适用于协议栈，不应该修改它。

功能说明

操作规程

典型应用代码包含三个单独阶段：初始化、普通操作和低功耗操作。



初始化后，组件将进入普通操作模式，并会定期进入各种低功耗操作模式，以保持电量。因此组件应该在系统上电时进行初始化，然后在正常操作模式和低功耗模式下运行。

系统初始化

初始化阶段发生在系统上电时或在从系统休眠唤醒时。该阶段会设置平台和组件参数。应用代码也应该启动该组件，并为将在其它他操作模式下发生的事件回调设置回调函数。

系统的正常操作

BLE_PDL 组件成功初始化或从休眠模式唤醒后，它将进入正常模式。如果尚未连接 BLE，那么正常操作模式将先建立与 BLE 的连接。然后，通过检查该协议栈的状态，它将处理所有挂起的 BLE 事件。通过调用 Cy_BLE_ProcessEvents()函数可以实现该操作。当处理完所有事件时，它将发送需要传输的数据。如果没有其它挂起事件，它将进入低功耗操作模式。在这种情况下，它将再次执行正常操作规程。在 BLE 连接事件周期内，至少要处理一次 BLE 事件。BLE 连接事件由中心器件在建立连接时进行配置。

系统的低功耗操作

在正常操作模式下没有挂起中断时，组件将进入低功耗模式。它先进入睡眠模式。组件可以进入休眠或深度休眠模式，具体情况取决于 BLE 接口硬件的状态。如果在低功耗模式期间发生了某个事件，它将再次进入正常操作模式。

请注意， MCU 和 BLE 子系统（BLESS）具有独立的功耗模式，可以相互独立进入不同的功耗模式。下表中各勾选标记表示 MCU 和 BLESS 的功耗模式的有效组合。

BLESS功耗模式	MCU功耗模式				
	活动模式	睡眠模式	深度睡眠模式	休眠模式	停止模式
活动模式（闲置/Tx/Rx）	✓	✓			
睡眠模式	✓	✓			
深度睡眠模式（关闭ECO）	✓	✓	✓		
禁止				✓	✓

器件绑定

与远程器件配对后，BLE_PDL 组件将存储所连接的链接密钥。如果该连接断开并重新建立，那么该器件将使用先前存储的密钥进行连接。

当这些器件被连接时，BLE 协议栈将更新 RAM 中的绑定数据。如果在关闭期间需要保留所绑定的数据，应用可以使用 Cy_BLE_StoreBondingData() API 将 RAM 中绑定的数据写入到专用的闪存

内，如该组件所定义。有关使用情况的详细内容，请参考 CE215121_BLE_HID_Keyboard 代码示例。

LFCLK 配置

设计范围资源（*<project>.cydwr*）文件 **Clocks** 选项卡中的 LFCLK 配置会影响组件在深度睡眠模式下的运行能力。如果选择了 WCO，则可以使用组件的深度睡眠模式。但如果选择的是 ILO，那么组件不会进入深度睡眠模式。

注意：只有在深度睡眠模式下，才能将 LFCLK 用于 BLE_PDL 组件，因此 ILO 的误差不会影响 BLE 通信。

多连接支持

BLE_PDL 在任何模式组合中最多支持同时进行四个多主多从（MMMS）BLE 连接。例如，它可以是四个从设备的主设备（4M）、三个从设备的主设备和另一设备的从设备（3M1S），或四个设备的从设备（4S），或任何其它组合。

要配置最大的 BLE 连接数量，请参阅 General 选项卡部分（**最大 BLE 连接数量**）。

BLE_PDL 组件支持 GATT 服务器的单个实例（单个 GATT 数据库）。在 BLE_PDL 配置对话框中，服务器实例数量字段始终被设置为 1。您可以将其它服务或完整的服务器配置文件添加到现有服务器树内并构建 GATT 数据库。这个单一的 GATT 数据库可以在所有 BLE 连接中重复使用。

BLE_PDL 组件管理多个 CCCD 值。每个有效连接的 CCCD 值是唯一的。组件支持的 CCCD 存储 SRAM 数据结构的最大数量取决于您所选择的有效 BLE 连接/链接的数量。组件与对端器件建立连接时，会对每个绑定器件恢复闪存中的 CCCD 值。

使用应用级的连接句柄来管理多连接。建立连接时（CY_BLE_EVT_ GATT_CONNECT_IND 事件的参数），将提供连接句柄（cy_stc_ble_conn_handle_t）。

为了使用特定连接，BLE API（BLE 协议栈 API 和 BLE 文件配置 API）提供参数 connHandle（例如，Cy_BLE_BASS_SendNotification（cy_stc_ble_conn_handle_t connHandle ...））。

来自 AppCallback 函数的 BLE 事件会在伴随事件产生的 eventParam 结构体中提供 connHandle，用于区分同该事件相关的连接。

以下代码显示了应用如何管理连接句柄：

```
/**
 * Allocate the connection handle array.
 * Macro CY_BLE_CONN_COUNT indicates the MAX number of supported connection.
 */

cy_stc_ble_conn_handle_t appConnHandle[CY_BLE_CONN_COUNT] =
{CY_BLE_INVALID_CONN_HANDLE_VALUE};

/* The callback event function to handle various events from the BLE stack */
void AppCallback (uint32 event, void* eventParam)
```

```
{
      switch (event)
       {
                    ...
            case CY_BLE_EVT_GATT_CONNECT_IND:
            /* Add the connected device to the connection handle array */
            appConnHandle[ (*(cy_stc_ble_conn_handle_t *)eventParam).attId ] =
                           *(cy_stc_ble_conn_handle_t *) eventParam;
            break;

            case CY_BLE_EVT_GATT_DISCONNECT_IND:
            /* Remove the connected device from the connection handle array */
            memset(&appConnHandle[(*(cy_stc_ble_conn_handle_t*)eventParam).attId],
                   CY_BLE_INVALID_CONN_HANDLE_VALUE,
                   sizeof(cy_stc_ble_conn_handle_t));
            break;
       }
}
```

为所有连接的器件实现循环：

```
for (i = 0; i < CY_BLE_CONN_COUNT; i++)
{
    if(Cy_BLE_GetConnectionState(appConnHandle[i]) == CY_BLE_CONN_STATE_CONNECTED)
    {
     /* Do some action */
    }
}
```

使用 Cy_BLE_GetDeviceRole（cy_stc_ble_conn_handle_t * connHandle）函数来发现链路层器件通过 connHandle 参数所指定的连接句柄连接到对端器件所使用的模式。

来自对端器件的写入属性（特征，描述符）请求，用于由 BLE 组件处理的所采用的服务（例如，BAS、HIDS、HRS 等）。

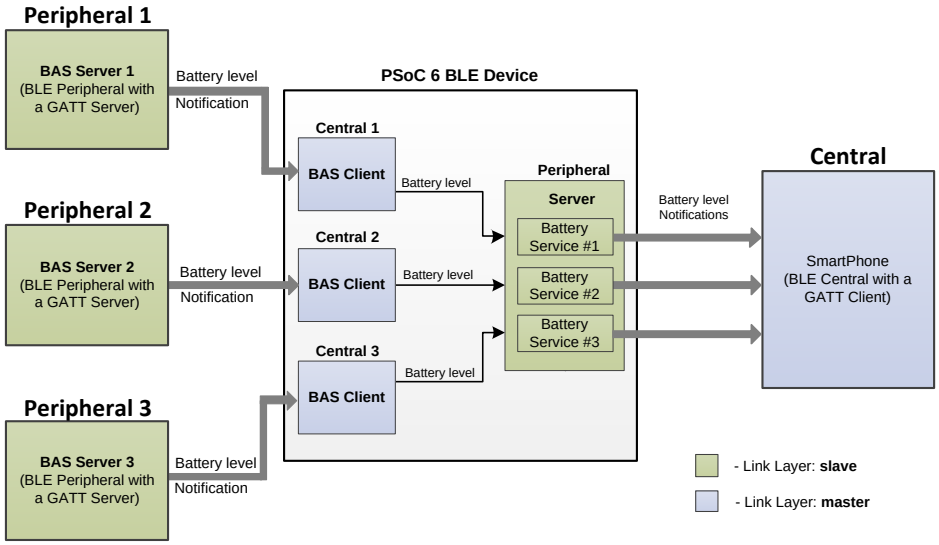
对于自定义服务，请在 AppCallback ()回调事件函数中编写应用级处理的属性请求。

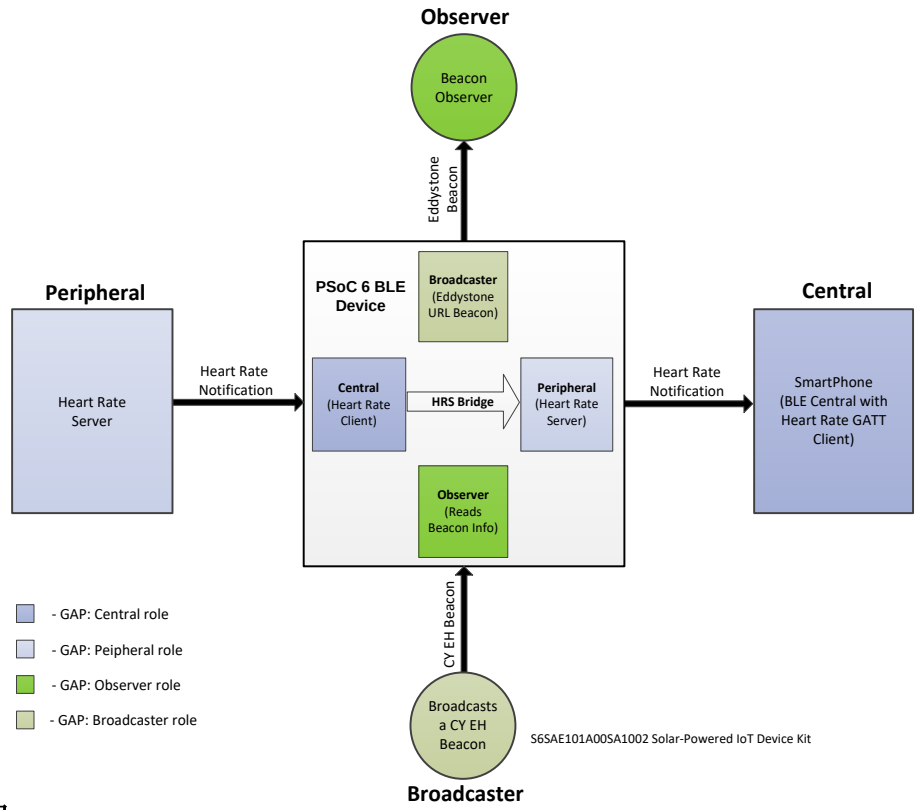
以下代码显示了对端器件对自定义服务的所进行的处理：

```
/* Call the back event function to handle various events from the BLE Stack */
void AppCallback (uint32 event, void* eventParam)
{
    switch (event)
    {
    ...
        case CY_BLE_EVT_GATTS_WRITE_REQ:
        {
            cy_en_ble_gatt_err_code_t gattErr;
            cy_stc_ble_gatt_write_param_t *writeParam =
                (cy_stc_ble_gatt_write_param_t *)eventParam;

            /* Store data in the database */
            gattErr = Cy_BLE_GATTS_WriteAttributeValuePeer(
                       &writeParam->connHandle, &writeParam->handleValPair);

            if(gattErr != CY_BLE_GATT_ERR_NONE)
            {
              /* Send an Error Response */
               cy_stc_ble_gatt_err_info_t errInfo =
                 {
                     .opCode     = CY_BLE_GATT_WRITE_REQ,
                     .attrHandle = writeParam->handleValPair.attrHandle,
                     .errorCode  = gattErr
                 };
                 Cy_BLE_GATTS_SendErrorRsp(&writeParam->connHandle, &errInfo);
            }
     ...
    }
}
```

在所有 GAP 模式（中心、外设、观察器和广播器）中配置 BLE_PDL 组件时，常见的 MMMS 使用了多主单从和多功能。

多主单从机使用框图





多功能使用框图

更多详细信息，请参考 CE215118_BLE_Multi_Master_Single_Slave 和 CE215555_BLE_Multi_Role 代码示例。

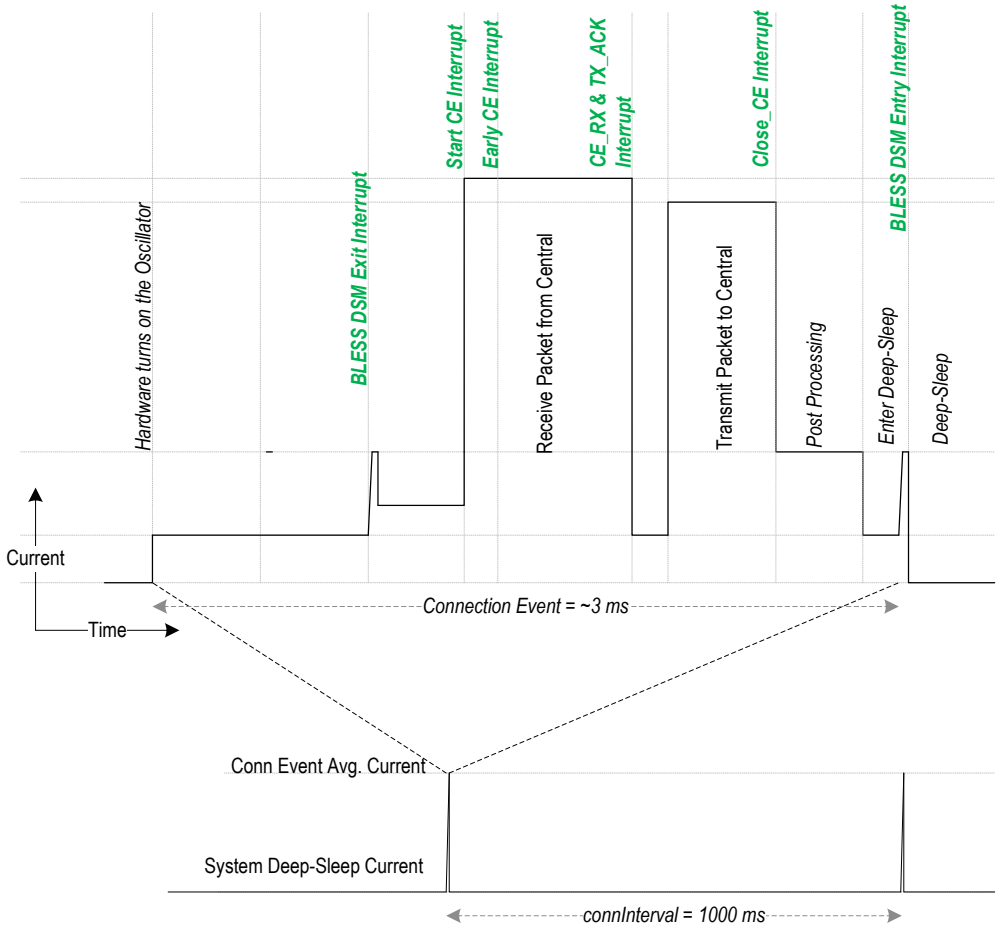
BLE 中断通知回调

BLE 组件向应用传递 BLE 中断通知，为用户指示不同的链路层和无线电状态转换（在发生 BLESS 中断的情况下）。

用户注册特定类型的回调，BLE 组件将根据注册的掩码调用已注册的回调（请参阅 Cy_BLE_RegisterInterruptCallback()和 Cy_BLE_UnRegisterInterruptCallback() API）。所有中断掩码都在 cy_en_ble_interrupt_callback_feature_t 枚举中指定。可能会触发用户回调的中断为：

1. **CY_BLE_INTR_CALLBACK_BLESS_STACK_ISR** — 在 BLESS 中断的每个触发器上执行。
2. **CY_BLE_INTR_CALLBACK_BLESS_INTR_STAT_DSM_EXITED** — 当 BLESS 退出深度睡眠模式并进入活动模式时执行。BLESS 退出深度睡眠模式的事件可以由需要 BLESS 处于活动状态的链路层硬件或不同的 BLE_PDL 数据传输 API 自动触发。
3. **CY_BLE_INTR_CALLBACK_BLELL_CONN_EXT_INTR_EARLY** — 当在从设备模式下的 BLESS 连接引擎检测到与其访问地址匹配的 BLE 数据包时执行。
4. **CY_BLE_INTR_CALLBACK_BLELL_CONN_INTR_CE_RX** — 当 BLESS 连接引擎接收到来自对端器件的非空数据包时执行。
5. **CY_BLE_INTR_CALLBACK_BLELL_CONN_INTR_CE_TX_ACK** — 当 BLESS 连接引擎从对端器件接收到之前发送的数据包的 ACK 数据包时执行。
6. **CY_BLE_INTR_CALLBACK_BLELL_CONN_INTR_CLOSE_CE** — 当 BLESS 连接引擎关闭连接事件时执行。无论数据的 tx/rx 状态如何，都会在每次连接间隔时执行该中断。

7. **CY_BLE_INTR_CALLBACK_BLESS_INTR_STAT_DSM_ENTERED** — 当 BLESS 进入深度睡眠模式时执行。用户对 Cy_SysPm_DeepSleep()函数的调用将触发 BLESS 进入深度睡眠的序列。在连接事件中触发每个中断（#1 到#7）的时间实例如下所示（绿色）。



8. **CY_BLE_INTR_CALLBACK_BLELL_SCAN_INTR_ADV_RX** — 当 BLESS 扫描引擎接收到来自对端器件的广播数据包时执行。
9. **CY_BLE_INTR_CALLBACK_BLELL_SCAN_INTR_SCAN_RSP_RX** – 当 BLESS 扫描引擎接收到来自对端设备的扫描响应数据包时（作为扫描器扫描请求的响应）将执行。
10. **CY_BLE_INTR_CALLBACK_BLELL_ADV_INTR_CONN_REQ_RX** — 当 BLESS 广播引擎接收到来自对端中心器件的连接请求时执行。

应用可以使用这些中断回调得知 RF 活动开始/结束、BLE 器件从广播状态改为已连接状态或者 BLESS 在活动模式和低功耗模式之间转换等事件。通过这些 BLESS 实时状态，可使应用与 BLESS 同步或防止与其它外设的无线电干扰等。

通过在 *cy_ble_config.h* 中定义 CY_BLE_INTR_NOTIFY_FEATURE_ENABLE，可以启用该功能。

```
#define CY_BLE_INTR_NOTIFY_FEATURE_ENABLE (1u) /* 1u - Enable / 0u – Disable */
```

BLE 双模式需要额外定义 IPC 通道和 IPC 中断结构，用于将通知从控制器内核发送给主机内核。使用以下定义：

```
#define CY_BLE_INTR_NOTIFY_IPC_CHAN      (15u) /* valid range:9..15, default:15 */
#define CY_BLE_INTR_NOTIFY_IPC_INTR      (15u) /* valid range:9..15, default:15 */
#define CY_BLE_INTR_NOTIFY_IPC_INTR_PRIOR (1u)  /* valid range:0..7,  default:1 */
```

不受支持的特性

BLE_PDL 组件协议栈不支持以下可选的蓝牙版本 5.0 协议功能，如规范的第 6 卷、B 部分、第 4.6 节所列内容：

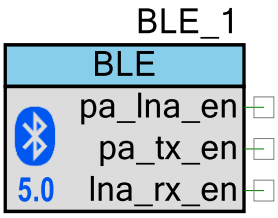
- 连接参数请求程序（第 6 卷、B 部分、第 4.6.2 节）
- 扩展的拒绝指示（第 6 卷、B 部分、第 4.6.3 节）
- 从设备发起的特性交换（第 6 卷、B 部分、第 4.6.4 节）
- 稳定调制指数 - 发送器（第 6 卷、B 部分、第 4.6.10 节）
- 稳定调制指数 - 接收器（第 6 卷、B 部分、第 4.6.11 节）
- 扩展型的低功耗广播（第 6 卷、B 部分、第 4.6.12 节）
- 定期低功耗广播（第 6 卷、B 部分、第 4.6.13 节）
- 第二个通道选择算法（第 6 卷、B 部分、第 4.6.14 节）
- 最小使用通道数程序（第 6 卷、B 部分、第 4.6.15 节）

低功耗模式

BLE 中间件自动注册睡眠和深度睡眠回调函数。该功能向 BLE 协议栈请求将低功耗蓝牙子系统（BLESS）置于低功耗模式。

输入/输出连接

本节介绍了 BLE 的输入和输出连接。I/O 列表中的星号（*）表示，在 I/O 说明部分中所列出的特定条件下，该 I/O 可能不可见。



pa_lna_en — 输出 *

通过该信号可以在没有无线电活动时使前端进入睡眠状态或待机状态。当 PA 控制或 LNA 控制为 ON 时，该信号为 ON。

该信号的极性是可配置的，可以通过 Cy_BLE_ConfigureExtPA() API 在 EXT_PA_LNA_CTRL 寄存器设置。

如果选中了 **Advanced** 选项卡中的 **nable external Power Amplifier (PA)**或 **Low Noise Amplifier (LNA) chip enable control** 参数，则该输出可见。

pa_tx_en — 输出 *

在传输期间该信号为打开状态，在不传输时转为关闭状态。实际开始传输之前该信号已经有效，这样可以为功率放大器提供上升所需时间。可以在 EXT_PA_LNA_DLY_CNFG 寄存器中设置该延迟。

该信号的极性是可配置的，可以通过 Cy_BLE_ConfigureExtPA() API 在 EXT_PA_LNA_CTRL 寄存器中设置。

如果在 **Advanced** 选项卡中选择了 **Enable external PA Tx control output** 参数，该输出便可见。

lna_rx_en — 输出 *

通过该信号可以在旁路路径和 LNA 路径之间进行选择。在接收过程中，该信号为 ON，而在接收器被关闭时转为 OFF。

该信号的极性是可配置的，可以通过 Cy_BLE_ConfigureExtPA() API 在 EXT_PA_LNA_CTRL 寄存器中设置。

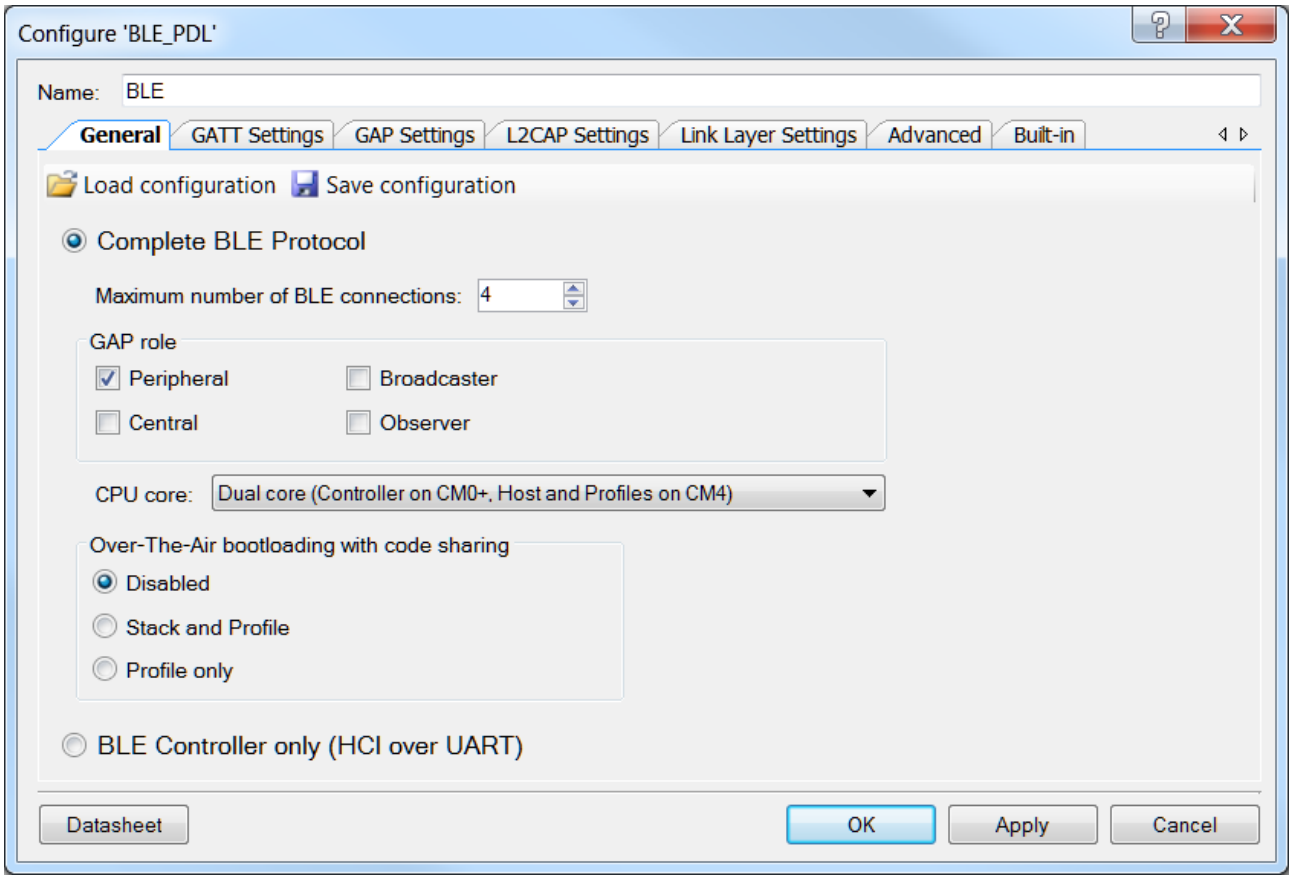
如果在 **Advanced** 选项卡中选择了 **Enable external PA Rx control output** 参数，该输出便可见。

组件参数

将 BLE_PDL 组件拖放到您的设计上，并双击以打开 Configure 对话框。该对话框包含下列各选项卡和参数。

General 选项卡

通过 **General** 选项卡可以对 BLE_PDL 组件进行常规配置。该选项卡包含用于加载和保存配置的工具以及配置类型的三个主要区域。



Load Configuration/Save Configuration 按键

使用 **Load Configuration** 按键加载先前保存的 xml 组件配置；使用 Save Configuration 按键保存当前的配置，供其它设计使用。能够以 xml 格式导入/导出定制器配置。

注意：为了以 **Bluetooth Developer Studio** 兼容格式加载或保存配置文件，请在 **GATT Settings** 选项卡上使用 **Load BDS Profile** 和 **Save Profile in BDS format** 工具类指令。

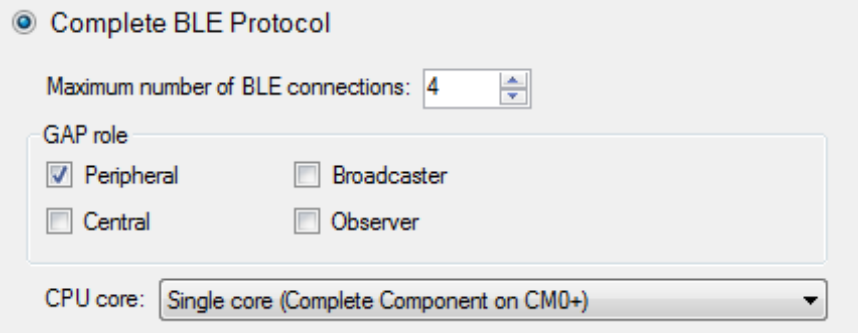
Mode Selection（模式选择）项

该选项卡的主要部分，有两个选项可供选择：

- Complete BLE Protocol（完整的 BLE 协议）项
- BLE Controller Only (HCI over UART)（仅限于 BLE 控制器）（使用 UART 接口的 HCI）项

General 选项卡 — Complete BLE Protocol 项

该项可用于启用 BLE 主机和控制器。所有 GAP 功能都已展示，供配置使用。



Maximum number of BLE connections（最多 BLE 连接数）项

该参数显示了所允许的 BLE 连接数量（包括中心器件和外设）。有效范围为 1 到 4。详细信息，请参阅多连接支持部分。

Gap Role（GAP 模式）

GAP role 参数可以采用以下各值：

- **Peripheral**（外设） — 定义了一个使用可连接的广播数据包进行广播的器件，连接后，该器件作为从设备。外设器件需要一个中心器件，因为中心器件可以启动连接。通过广播数据，外设器件可以广播有关器件的普通信息。
- **Central**（中心器件） — 定义了一个启动外设连接的器件，连接后，该器件作为主设备。外设器件需要一个中心器件，因为中心器件可以启动连接。
- **Broadcaster**（广播器） — 与外设模式相同，器件将发送广播数据。但广播器不支持连接，它只能发送数据，却不会接收数据。
- **Observer**（观察器） — 在该模式下，器件对广播器进行扫描，并向应用报告所接收的信息。观察器模式不允许传输数据。

CPU Core（CPU 内核）

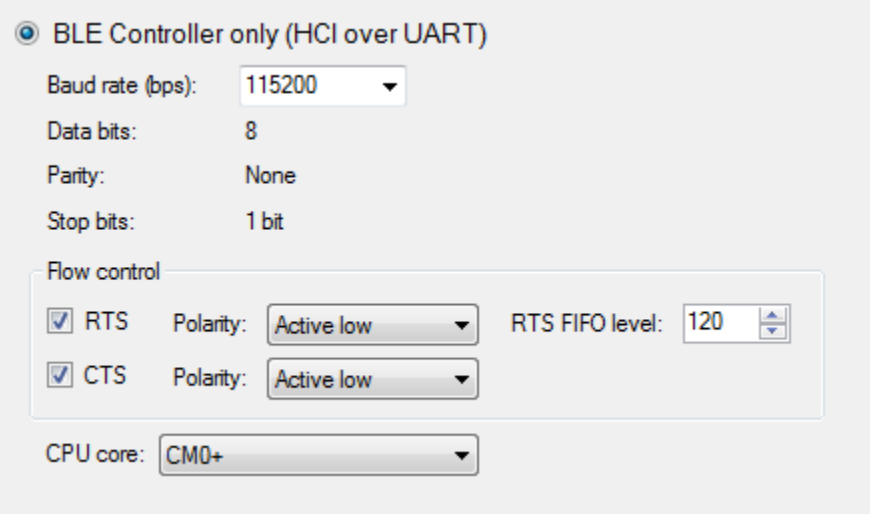
CPU Core 参数用于定义内核的使用。可以将其设置为以下各值：

- **Single core (Complete Component on CM0+)** — 只会使用 CM0+内核。
- **Single core (Complete Component on CM4)** — 只会使用 CM4 内核。

  注意：对于单核 MPNs，只能选择 **Single core (Complete Component on CM4)**项。
- **Dual core (Controller on CM0+, Host and Profiles on CM4)** — 可以使用两个内核：CM0+用于控制器，CM4 用于主机和配置文件。

General 选项卡 — BLE Controller only (HCI over UART)项

选择该配置后可使组件进入 HCI 模式，从而能够将器件作为 BLE 控制器使用。这样还能够使器件通过组件嵌入式 UART 与主机协议栈进行通信。选择该模式时，**GATT Settings** 选项卡、 **GAP Settings** 选项卡以及 **L2CAP Settings** 选项卡不可用。



UART 包括一个全双工 8 数据位、1 个停止位、无奇偶校验位、无流量控制接口。

- **波特率（bps） —** 配置 UART 波特率。
- **RTS —** 通过该参数可以使能 Ready to Send （RTS）输出信号。RTS 信号是接收器所使用的流控制功能的一部分。只要接收器准备好接收更多数据，它将保持 RTS 信号的有效状态。RTS FIFO 电平参数决定了 RTS 是否有效。默认值为：True。
- **RTS Polarity**（RTS 极性） **—** 该参数决定 RTS 输出信号的有效极性为低电平有效（默认设置）还是高电平有效。

- **RTS FIFO level**（RTS FIFO 电平） **—** 该参数决定了 RTS 是否有效。RX FIFO 中的输入项小于 RTS FIFO 电平参数指定的值时，RTS 信号将保持有效状态；否则，RTS 信号会转为无效状态。从读取 RX FIFO 中的单位数据开始，RTS 一直保持无效状态，直到匹配 RTS FIFO 电平为止。默认值为：120。
- **CTS —** 通过该参数，可以使能路由到引脚上的“Clear to Send”（CTS）输入信号。CTS 信号是发送器所使用的流控制功能的一部分。发送器在发送 TX FIFO 中的数据前将检查 CTS 信号是否处于有效状态。如果 CTS 信号为无效状态，则数据传输将被挂起，并且在 CTS 信号转为有效状态时恢复数据传输。默认值为：True。
- **CTS Polarity**（CTS 极性） **—** 该参数决定 CTS 输入信号的有效极性为低电平有效（默认设置）还是高电平有效。
- **CPU Core**（CPU 内核） **—** 该参数用于定义内核的使用：CM0+还是 CM4。

通过代码共享特性进行无线引导加载

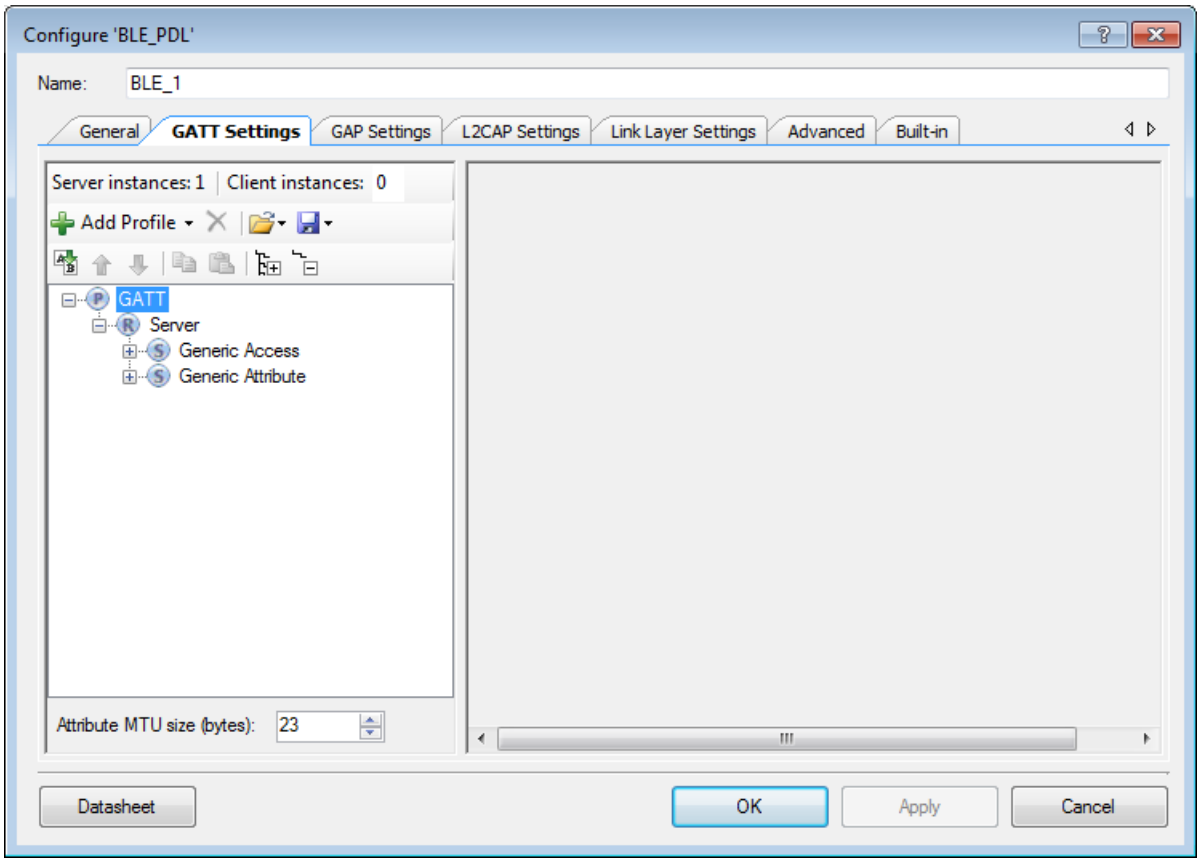
该选项用于无线（OTA）实现。通过该选项，您可以在两个组件实例之间共享 BLE 组件代码：一个实例具有特定于配置文件的代码，另一个实例具有协议栈。该参数用于选择以下选项：

- **Disabled**（禁用） — 该选项禁用 OTA 功能。
- **Stack and Profiles**（协议栈和配置文件） — 选择该选项时，组件仅表示具有强制性 Bootloader 服务和其它用户首选服务的 BLE 的协议栈部分。该组件选项用于隔离协议栈和应用配置文件。选择该选项后，Profiles 选项卡将使用 BLE Bootloader 服务自动填充，用户能够在 Bootloader 服务之上添加其它服务。

  注意： 该模式需要占用大约 3024 个额外字节的堆内存。如果没有足够的堆内存，BLE 组件将无法工作。可以通过编辑链接器脚本来修改堆大小。
- **Profile only**（仅限配置文件） — 该选项使组件仅具有特定于配置文件的代码。协议栈被排除在外。

GATT Settings 选项卡

GATT Settings 选项卡用于配置“配置文件”特定的参数。它直接受 **General** 选项卡中的 **Profile** 设置选择的影响。**GATT Settings** 选项卡包括三个区域：工具栏、配置文件树和参数配置域。



工具栏

工具栏包含导航选项以及添加或删除服务、特性和描述符的方式。

- **Server instances**（服务器实例） — 该参数指定 GATT 服务器实例的数量。BLE_PDL 组件支持 GATT 服务器的单个实例（单个 GATT 数据库）。您可以将其它服务或完整的配置文件添加到现有服务器配置文件树并构建 GATT 数据库。这个单一 GATT 数据库可以在所有 BLE 连接中重复使用。

  注意：每个有效连接的 CCCD 值将是唯一的。

- **Client instances**（客户端实例） — 该参数指定 GATT 客户端实例的数量。每个连接存在一个 GATT 客户端实例。您最多可以配置四个 GATT 客户端实例。所有 GATT 客户端实例都有一个通用的客户端配置文件树配置。

- **Add Profile**（添加配置文件） — 当 GATT 节点在配置文件树中高亮显示时，该选项是可用的。通过它，可以将整个配置文件添加到配置文件树中。此选项不会删除树中的现有服务。文件树中可以同时存在多个配置文件。
- **Add Service**（添加服务） — 当 **Profile Role** 在配置文件树中高亮显示时，该选项是可用的。它允许在选定的 **Profile Role** 中加载服务。在 GATT 服务器配置中，使用该选项可将选定的服务数据添加到服务器 GATT 数据库内，并使能特定服务 API。在 GATT 客户端配置中，用于该服务的自动检测功能的数据结构是由组件创建的。如果在自动检测过程中发现了 GUI 中不存在服务，那么组件会忽略这些服务，并且应用程序将负责查找这些服务的详细信息。有关可用的服务的信息，请参考 Profile 内容。
- **Add Characteristic**（添加特性） — 当在配置文件树中高亮显示某个服务时，则表明该选项是可用的。特性选项是每个服务独有的，如果服务被添加到设计中，将会自动加载所有特性选项。**Add Characteristic** 按键可用于手动添加服务的新特性。上面提到的服务所具有的全部特性和自定义特性均是可选的。
- **Add Descriptor**（添加描述符） — 当在配置文件树中高亮显示某个特性时，表示该选项是可用的。与特性选项相同，Descriptor（描述符）是特性独有的选项，如果该特性被添加到设计中，将自动加载所有描述符选项。更多有关 BLE 特性描述符的信息，请访问 developer.bluetooth.org。（**请注意，**要想访问该网站，您必须是蓝牙 SIG 的成员。）
- **Delete**（删除） — 删除已选定的服务、特性或描述符。
- **Load/Save**（加载/保存） — 导入/导出该树中所示的配置文件、服务、特性和描述符。该功能独立于 **General** 选项卡上的 **Load Configuration/Save Configuration**（加载配置/保存配置）按键。因此您可以自定义这个文件树而不会影响到通用设置。每个导出文件类型都有自己的扩展内容。

  BLE_PDL 组件支持以 **Bluetooth Developer Studio** 工具的文件格式导入和导出配置文件。使用 **Load BDS Profile** 命令导入 BDS 配置文件，并通过 **Save Profile in BDS format** 命令以 BDS 文件格式导出配置文件。
- **Rename**（重命名） — 重命名 Profiles tree（配置文件树）中的选定项。
- **Move Up/Down**（上移/下移） — 在配置文件树中上移或下移所选选项。
- **Copy/Paste**（复制/粘贴） — 用于复制/粘贴配置文件树中的各项。
- **Expand All**（扩展全部） — 展开配置文件树中的所有项。
- **Collapse all Services**（折叠所有服务） — 折叠起配置文件树中的所有项。

配置文件树

配置文件树用于查看GATT服务器和客户端模式下的GATT服务、特性和描述符。通过浏览该树，您可以使用工具栏按键或上下文菜单快速添加、删除或修改服务、特性和描述符。您可以通过点击该树中的某一项来配置参数。这些参数会显示在参数配置域中。

参数配置

参数配置域允许您通过选择树中的服务或特性类型对配置文件、服务或特性进行配置。

属性 MTU 大小

设计中使用了属性的最大传输单元大小（字节）。有效范围为 23 至 512 字节。该值用于响应来自 GATT 客户端的交换 MTU 请求。

配置文件

您可以将整个配置文件从受支持的配置文件列表中添加到配置文件树中。可以选择以下配置文件：

警报通知

该配置文件允许 GATT 客户端接收各种类型的警报和事件信息，以及 GATT 服务器器件中有关新警报和未读取项目的计数信息。

- **Alert Notification Server**（警报通知服务器）配置文件模式 — 被指定为 GATT 服务器。需要以下服务：**警报通知服务**。
- **Alert Notification Client**（警报通知客户端）配置文件角色 — 被指定为 GATT 客户端。

要想了解更多有关警报通知配置文件的信息，请查阅警报通知配置文件规范的内容。

自动化 IO

通过该配置文件，器件能够连接一个自动化 IO 模块（IOM），并与之交互，从而可以访问数字和模拟信号。

- **自动化 IO 服务器**配置文件模式 — 被指定为 GATT 服务器。需要以下服务：**自动化 IO 服务**。
- **自动化 IO 客户端**配置文件模式 — 被指定为 GATT 客户端。

要想了解更多有关自动化 IO 配置文件的信息，请查阅自动化 IO 配置文件规范的内容。

血压配置文件

该配置文件允许器件与血压传感器设备相互连接并进行通信，这样能够使用在消费类和专业医疗应用中。

- **血压传感器**配置文件模式 — 被指定为 GATT 服务器。需要以下服务：**血压服务、器件信息服务**。
- **血压收集器**配置文件角色 — 被指定为 GATT 客户端。需要支持以下服务：**血压服务**。可以选择支持**器件信息服务。**

有关血压配置文件的详细信息，请参阅血压配置文件规范内容。

连续血糖监测

该配置文件允许器件与连续血糖监测传感器相连并进行通信，这样能够应用在消费类的医疗应用中。

- **连续血糖监测传感器**配置文件模式 — 被指定为 GATT 服务器。需要以下服务：**连续血糖监测服务、器件信息服务**。可选择包括**绑定管理服务**。
- **收集器**配置文件模式 — 被指定为 GATT 客户端。需要支持以下服务：**连续血糖监测服务**。可选择支持**绑定服务**和**器件信息服务。**

有关连续血糖监测配置文件的详细信息，请参阅连续血糖监测配置文件规范内容。

循环供电

该配置文件允许收集器与循环供电传感器相连并通信，这样可以使用在体育和健身应用中。

- **循环供电传感器**配置文件模式 — 被指定为 GATT 服务器。需要以下服务：**循环供电服务**。可能需要包括**器件信息服务**和**电池服务**。
- **循环供电传感器和广播器**配置文件模式。需要以下服务：**循环供电服务**。
- **收集器**配置文件模式 — 被指定为 GATT 客户端。需要支持以下服务：**循环供电服务**。可选择支持**器件信息服务**和**电池服务。**
- **循环供电观察器**配置文件模式。只能在**循环供电广播器**模式**下与器件通信。**

请参考循环供电配置文件规范内容，了解更加详细的信息。

循环速度和节奏

该配置文件可使收集器与循环速度和节奏传感器相连并通信，可使用在体育和健身应用中。

- **循环速度和节奏传感器**配置文件模式 — 被指定为 GATT 服务器。需要以下服务：**循环速度和节奏服务**。可选择支持**器件信息服务**。
- **收集器**配置文件模式 — 被指定为 GATT 客户端。需要支持以下服务：**循环速度和节奏服务**。可以选择支持**器件信息服务。**

请参考循环速度和节奏配置文件规范，了解更加详细的信息。

环境感应概况

该配置文件能够使收集器与环境传感器建立连接并通信，这样可以使用于户外活动应用中。

- **环境传感器**配置文件模式 — 被指定为 GATT 服务器。需要以下服务：**环境感应服务**。可以选择包括**器件信息服务**和**电池服务**。
- **收集器**配置文件模式 — 被指定为 GATT 客户端。需要支持以下服务：**环境感应服务**。可选择支持**器件信息服务**和**电池服务。**

请参考环境感应配置文件规范内容，了解更加详细的信息。

Find Me（查找）配置文件

查找配置文件定义了按下设备上按键时的特性，以便在对端设备上生成一个警报信息。

- **查找目标**配置文件模式 — 被指定为 GATT 服务器。需要以下服务：**即时警报服务**。
- **查找定位器**配置文件模式 — 被指定为 GATT 客户端。需要支持以下服务：**即时警报服务**。

欲了解更详细的信息，请参考查找配置文件规范部分。

血糖

该配置文件允许器件与血糖传感器相连并进行通信，这样能够应用在消费类的医疗应用中。

- **血糖传感器**配置文件模式 — 被指定为 GATT 服务器。需要以下服务：**血糖服务、器件信息服务**。

- **收集器**配置文件模式 — 被指定为 GATT 客户端。需要支持以下服务：**血糖服务**。可以选择支持**器件信息服务**。

欲了解更详细的信息，请参考查找配置文件规范部分。

健康温度计

该配置文件允许器件与血糖传感器相连并进行通信，这样可以使用在消费类的医疗应用中。

- **温度计**配置文件模式 — 被指定为 GATT 服务器。需要以下服务：**健康温度计服务**、**器件信息服务**。
- **收集器**配置文件模式 — 被指定为 GATT 客户端。需要支持以下服务：**健康温度计服务**。可以选择支持**器件信息服务**。

更多有关信息，请参见健康温度计配置文件规范内容。

HTTP 代理

该服务允许客户端器件（通常是传感器）通过网关器件与网络服务器通信。**Add Profile**（添加配置文件）下拉列表中不提供 HTTP 代理服务。可以单独添加该服务。

有关 HTTP 代理服务的详细信息，请参阅 HTTP 代理服务规范内容。

心率

该配置文件能够使收集器与心率传感器相连并进行通信，这样可以使用于健身应用。

- **心率传感器**配置文件模式 — 被指定为 GATT 服务器。需要以下服务：**心率服务**、**器件信息服务**。
- **收集器**配置文件模式 — 被指定为 GATT 客户端。需要支持以下服务：**心率服务**。可以选择支持**器件信息服务**。

欲了解更详细的信息，请参考心率配置文件规范部分。

GATT 上的 HID

该配置文件定义了一个具有 BLE 无线通信功能的器件如何使用通用属性来配置文件，使之在 BLE 协议协议栈上支持 HID 服务。

- **HID 器件**配置文件模式 — 被指定为 GATT 服务器。需要以下服务：**HID 服务、电池服务**以及**器件信息服务**。可以选择**扫描参数服务**作为**扫描参数**配置文件的**扫描服务器**模式的一部分。**HID 器件**支持 **HID 服务**和**电池服务**多个实例，并且可以包括任何其它可选服务。
- **引导主机**配置文件模式 — 被指定为 GATT 客户端。需要支持以下服务：**HID 服务**。可选择支持**电池服务**和**器件信息服务。**
- **上报主机**配置文件模式 — 被指定为 GATT 客户端。需要支持以下服务：**HID 服务、电池服务以及器件信息服务**。可选择支持**扫描参数的**扫描客户端**模式。**
- **上报和引导主机**配置文件 — 被指定为 GATT 客户端。需要支持以下服务：**HID 服务、电池服务以及器件信息服务**。可选择支持**扫描参数的**扫描客户端**模式。**

欲了解更详细的信息，请参考 GATT 上的 HID 配置文件规范部分。

室内定位

室内定位服务公开位置信息，这样即使在 GNSS 信号不可用的环境中移动设备仍能给出自己的定位。例如，在室内场所。位置信息主要通过广播公开，而基于 GATT 的服务主要用于配置。

Profile 下拉列表中不提供室内定位服务。可以单独添加该服务。

有关室内定位服务的详细信息，请参阅室内定位服务规范内容。

互联网协议支持

该配置文件支持在各器件间通过蓝牙低功耗传输交换 IPv6 数据包。IPSP 定义了两个模式，即节点模式和路由器模式。器件可以支持两个节点和路由器模式。支持节点模式的器件可能是传感器或执行器。支持路由器模式的器件可能是一个接入点（例如家庭路由器、移动电话或类似设备）。

- **节点**配置文件模式 — 被指定为 GATT 服务器。需要支持以下服务：**互联网协议支持服务**。
- **路由器**配置文件模式 — 被指定为 GATT 客户端。需要支持以下服务：**互联网协议支持服务**。

有关 IPSP 的详细信息，请参阅互联网协议支持配置文件规范内容。

定位和导航

该配置文件允许器件与定位和导航传感器建立连接并通信，这样可以使用于户外活动应用中。

- **定位和导航传感器**配置文件 — 被指定为 GATT 服务器。需要以下服务：**定位和导航服务**。可以选择包括**器件信息服务**和**电池服务**。

- **收集器**配置文件模式 — 被指定为 GATT 客户端。需要支持以下服务：**定位和导航服务**。可选择支持**器件信息服务**和**电池服务。**

更多有关信息，请参见定位和导航配置文件规范内容。

手机警报状态

该配置文件允许器件向用户报告连接至器件的手机警报状态。

- **手机警报服务器**配置文件模式 — 被指定为 GATT 服务器。需要以下服务：**手机警报状态服务**。
- **手机警报客户端**配置文件模式 — 被指定为 GATT 客户端。需要支持以下服务：**手机警报服务**。

更多有关信息，请参考手机警报状态配置文件规范内容。

接近感应

接近感应配置文件会使能两个器件间的接近感应监控。

- **接近感应报告**配置文件模式 — 被指定为 GATT 服务器。需要以下服务：**链路损耗服务**。如果**即时警报服务**和 **Tx 电源服务**都被使用，那么可使用这二者。不允许只使用其中一个可选服务。
- **接近感应监视器**配置文件模式 — 被指定为 GATT 客户端。需要支持以下服务：**链路损耗服务**。可选择支持**即时警报服务**和 **Tx 电源服务**。相同的限制也适用于**接近感应报告**。

欲了解更详细的信息，请参考接近感应配置文件规范部分。

脉搏血氧计

该配置文件允许器件与脉搏血氧计设备相互连接并进行通信，这样能够使用在消费类和专业医疗应用中。

- **脉搏血氧计传感器**配置文件模式 — 被指定为 GATT 服务器。需要以下服务：**脉搏血氧计服务、器件信息服务**。可选择包括绑定管理服务、当前时间服务以及电池服务。
- **收集器**配置文件模式 — 被指定为 GATT 客户端。需要以下服务：**脉搏血氧计服务**和**器件信息服务。** 可选择支持绑定管理服务、当前时间服务以及电池服务。

欲了解更详细的信息，请参考脉搏血氧计配置文件规范部分。

运行速度和节奏

该配置文件允许收集器与运行速度和节奏传感器相连并进行通信，这样可以使用于体育和健身应用中。

- **运行速度和节奏传感器**配置文件模式 — 被指定为 GATT 服务器。需要以下服务：**运行速度和节奏服务**。可选择支持**器件信息服务**。
- **收集器**配置文件模式 — 被指定为 GATT 客户端。需要支持以下服务：**运行速度和节奏服务**。可以选择支持**器件信息服务**。

请参考运行速度和节奏配置文件规范，了解更加详细的信息。

扫描参数

该配置文件定义了一个具有 BLE 无线通信功能的扫描客户端是如何将它的扫描特性写入到扫描服务器中的，并定义了一个扫描服务器是如何请求更新扫描客户端的扫描特性的。

- **扫描服务器**配置文件模式 — 被指定为 GATT 服务器。需要以下服务：**扫描参数服务**。
- **扫描客户端**配置文件模式 — 被指定为 GATT 客户端。需要支持以下服务：**扫描参数服务**。

更多有关信息，请参考扫描参数配置文件规范内容。

时间

时间配置文件允许器件获取有关日期、时间、时区和 DST 等方面的信息，并控制与时间相关的功能。

- **时间服务器**配置文件模式 — 被指定为 GATT 服务器。需要以下服务：**当前时间服务**。可选择支持**下一个 DST 更改服务**和**参考时间更新服务**。
- **时间客户端**配置文件模式 — 被指定为 GATT 客户端。需要支持以下服务：**当前时间服务**。可选择支持**下个 DST 更改服务**和**参考时间更新服务**。

欲了解更详细的信息，请参考时间配置文件规范部分。

计量秤

计量秤配置文件允许数据收集设备获取支持计重服务的计量秤中的数据。

- **计量秤**配置文件模式 — 被指定为 GATT 服务器，也可能是 GATT 客户端。

需要以下服务：**计量秤服务**和**器件信息服务**。

可以选择支持：**用户数据服务**、**身体组成服务**、**电池服务**以及**当前时间服务**。

- **收集器**配置文件模式 — 被指定为 GATT 客户端，也可以指定为 GATT 服务。

  需要支持以下服务：**计量秤服务**和**器件信息服务**。

  可以选择支持：**用户数据服务**、**身体组成服务**、**电池服务**以及**当前时间服务**。

欲了解更详细的信息，请参考计量秤配置文件规范部分。

无线电源传输

无线电源传输配置文件（A4WP）可实现无线电源传输系统中电源接收器单元和电源发送器单元之间的通信。

- **电源接收器单元**配置文件模式 — 被指定为 GATT 服务器。需要以下服务：**无线电源传输**。
- **电源发送器单元**配置文件模式 — 被指定为 GATT 客户端。需要支持以下服务：**无线电源传输**。

无线电源传输配置文件是由无线电源联盟（A4WP）定义的一个自定义服务。有关无线电源传输配置文件的详细信息，请访问 AirFuel 联盟网站。

Bootloader 配置文件

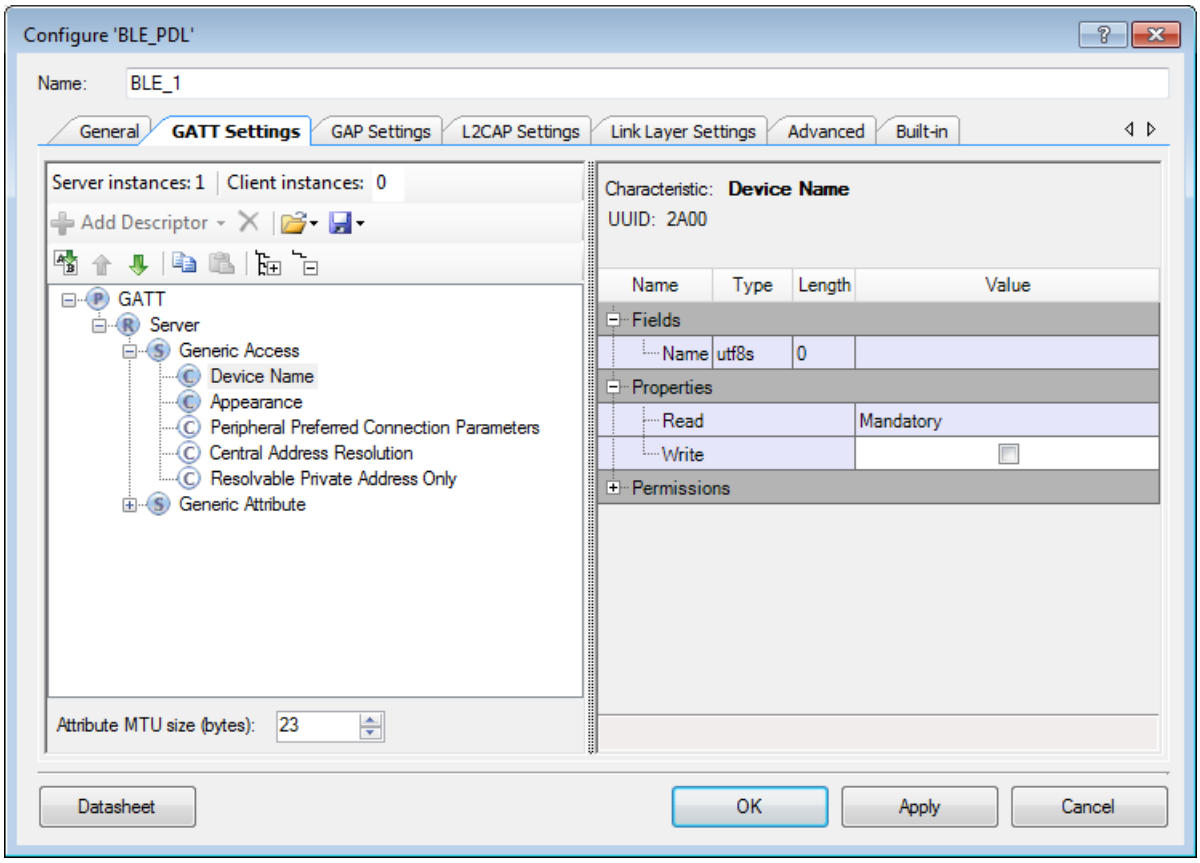
该组件支持 Bootloader 配置文件和 Bootloader 服务，它允许 Bootloader 组件更新赛普拉斯 BLE 设备上的现有固件。Bootloader 服务使用蓝牙低功耗接口作为通信接口。如果设计需要无线（OTA）更新固件，则可将它添加到任何配置文件中。

有关 Bootloader 服务的详细信息，请参阅 Bootloader 服务配置部分。

注意：

- 所有配置文件至少要有一个**通用访问服务**和**通用属性服务**。
- 只有服务特性属于一个 GATT 服务器节点时，才能配置它们。
- **GAP Settings**（GAP 设置）选项卡中的安全设置被全局化使用。另外，您可以手动配置每个特性/描述符的安全性。
- 树节点图标使用了两种颜色：蓝色和白色。蓝色表示节点是必需项，无法删除。白色表示节点是可选的。

通用接入服务



该服务用于定义蓝牙的基本连接，并检测各个参数。点击 **Generic Access Service** 下的特性，然后查看特殊特性的设置情况。然后，执行 **GAP Settings** 选项卡中 **General** 选项内的实际特性配置。

- **Device Name**（器件名称）：该参数用于指定器件名称。它默认拥有相应的读（未经验证/授权）和写属性。该参数最多可达 248 个字节。该值来自 GAP Settings 选项卡上 General 下的 **Device Name** 字段。
- **Appearance**（外观）：显示器件的徽标或外观（它是 SIG 定义的 2 字节值）。它默认拥有相应的读（未经验证/授权）和写属性。该值来自 GAP Settings 选项卡中 General 下的 **Appearance** 字段。

- **Peripheral Preferred Connection**（外设首先连接）：充当外设功能的器件可以将首选连接的参数传送到对端器件内。该参数共有 8 个字节，并由下面各子参数组成。

  请注意，仅当器件支持外设功能时，该参数才可用。更多有关信息，请参考 GAP Settings 选项卡的外设首选连接参数部分。

  - **Minimum Connection Interval**（最短连接间隔）：它是一个 2 字节的参数，表示允许连接的最短时间。
  - **Maximum Connection Interval**（最长连接间隔）：它是一个 2 字节的参数，表示允许连接的最长时间。
  - **Slave Latency**（从设备延迟）：这是一个 2 字节的数值，它定义了连续连接事件间的延迟。
  - **Connection Supervision Timeout Multiplier**（连接监控超时乘数）— 该参数是 2 字节的数值，表示 LE 链接监控超时间隔的大小。在对端设备没有通过 LE 链接发出响应的情况下，可通过该参数指定 LE 链接需要保持的超时时长。

    请注意，为了正常运行，连接监控超时必须大于**（1 +从设备延迟）*连接间隔* 2**（ms）。有关连接监控超时的更多信息，请参阅蓝牙内核规范第 6 卷、B 部分、第 4.5.2 章。

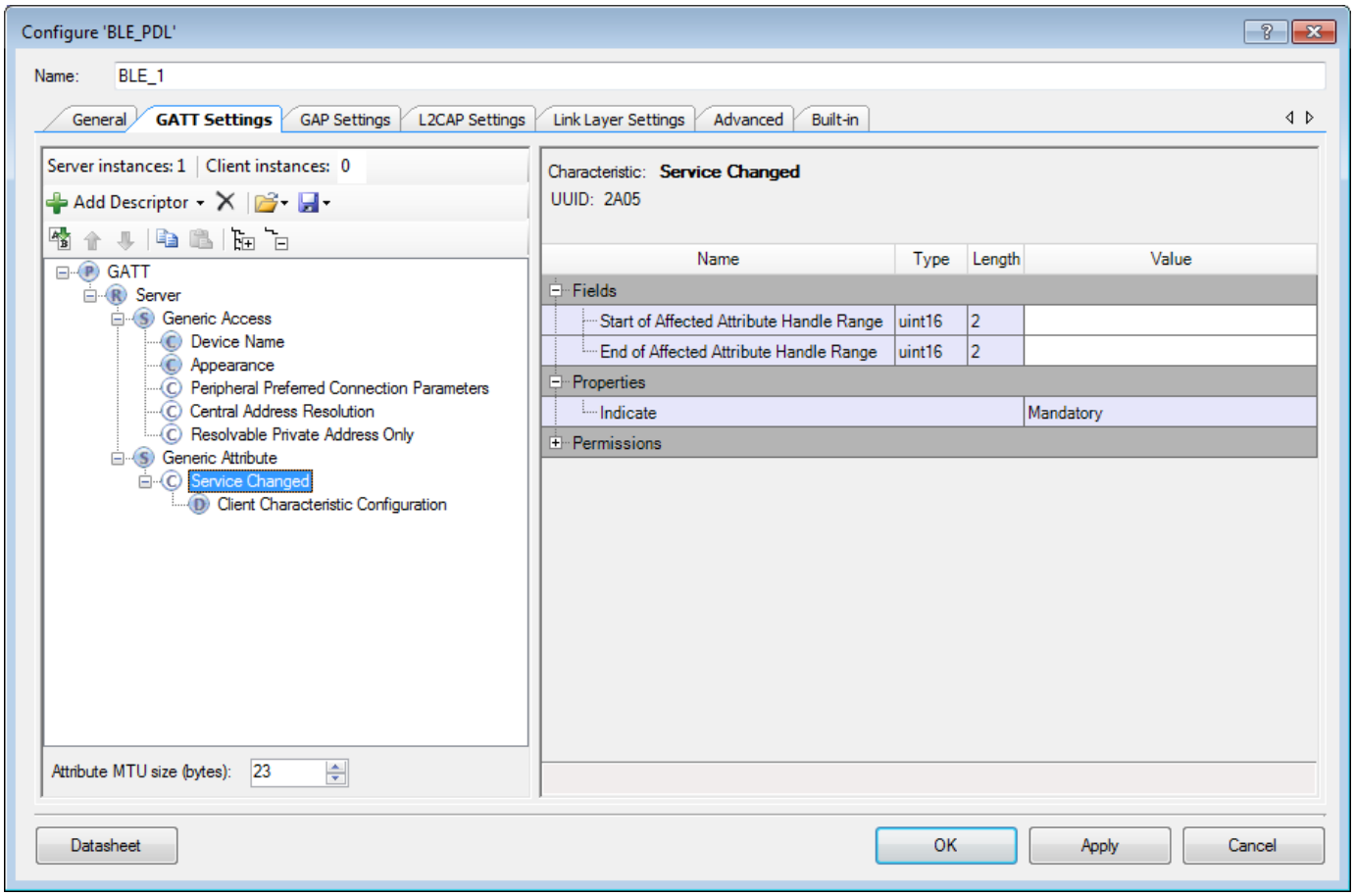
  注意：如果 GAP 中心器件没有使用“优先连接外设”参数，那么上述各参数用于通过 L2CAP 更新连接参数过程。例如，iOS7 忽略了“优先连接外设”参数，并建立了一个默认间隔为 30 ms 的连接。外设器件通过在合适的时间内发送 L2CAP 连接参数的更新请求，以此要求更新连接参数。

  如果周期性的通知或指示配置了某种特性，那么典型的外设将启动 L2CAP 连接参数的更新过程。

- **Central address resolution**（中心地址解析）：在中心模式下的器件可以通过地址解析来表示它是否支持隐私。在使用定向广播（其中发起方地址设置为可解析私有地址（RPA））之前，外设应通过读取中心地址解析特性来检查对端设备是否支持地址解析。

- **Resolvable Private Address Only**（仅限于可解析私有地址）：定义器件是否仅将可解析私有地址（RPA）用作本地地址。

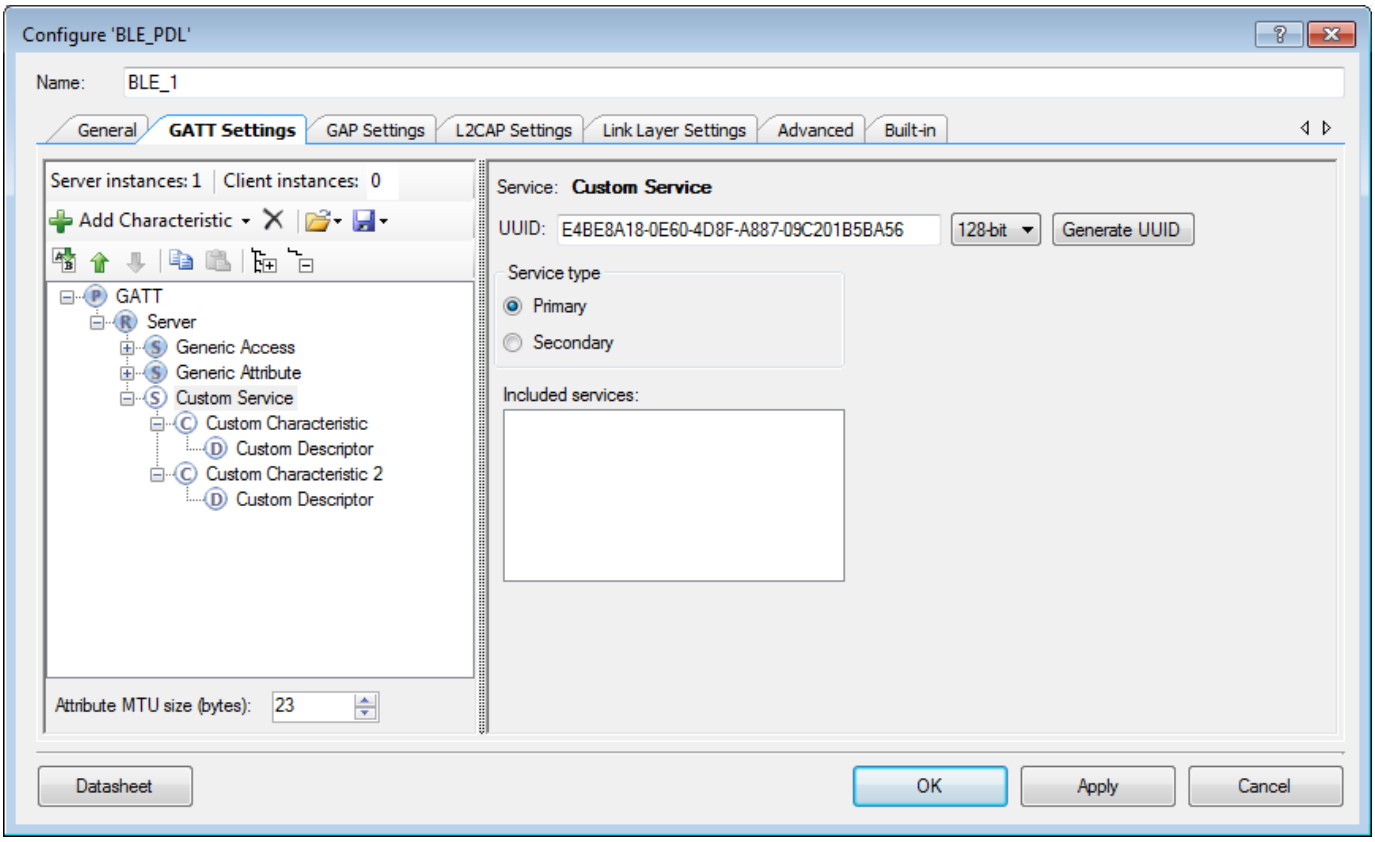
Generic Attribute Service（通用属性服务）



点击 Generic Attribute Service 下的 Characteristic，然后配置特定特性。

- **Service Changed**（变化的服务） — 通过该特性可以向连接的器件指出服务已经发生了变化（例如：添加、删除或修改）。当 GATT 客户端重新连接到 GATT 服务器时，该特性也会通知与 GATT 服务器有关系（比如：绑定）的 GATT 客户端：基于 GATT 的服务已经发生变化。对于充当 GATT 客户端模式的器件，该特性是必要的。对于充当 GATT 服务器模式的器件，如果 GATT 服务器更改了器件中受支持的服务，那么该特性是必要的。

自定义服务配置



UUID

服务的通用惟一标识符。该字段是自定义服务中可编辑的字段。使用 **Generate** 按键随机生成 128 位 UUID。

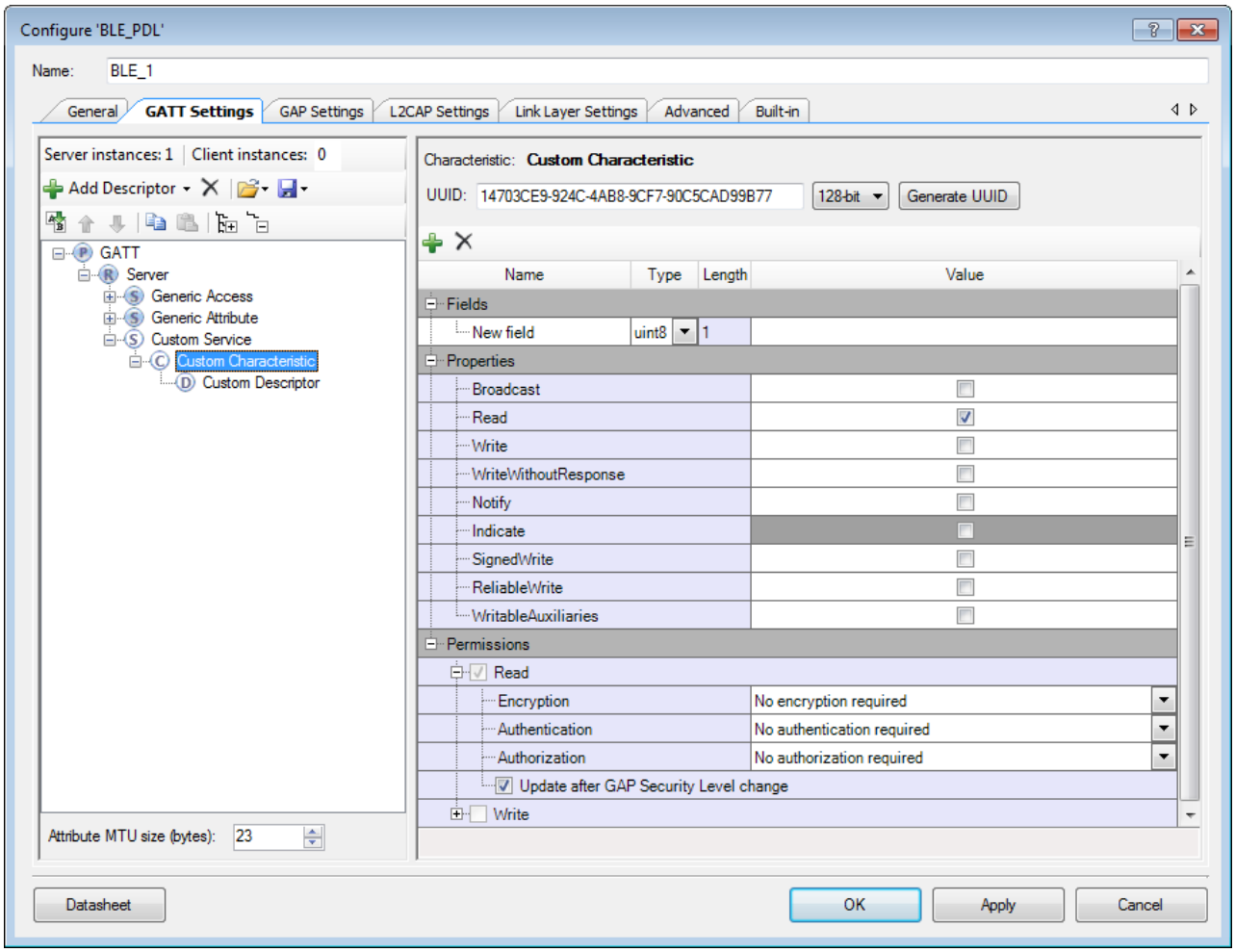
服务类型

- **Primary** — 表示器件的基本功能。
- **Secondary** — 表示器件的辅助功能。器件的辅助功能必须包含在另一个服务内。

所包含的服务

- 下表显示的是已选服务内可以包含的服务。每种服务能够包含一个或多个服务。所包含的服务为该服务提供辅助功能。

自定义特性配置



UUID

特性的惟一通用标识符。该字段是自定义特性中可编辑的字段。使用 **Generate** 按键随机生成 128 位 UUID。

Fields（字段）

字段代表一个特性值。可以在 **Value** 列中设置每个字段的默认值。在自定义特性的情况中，可以自定义该字段。

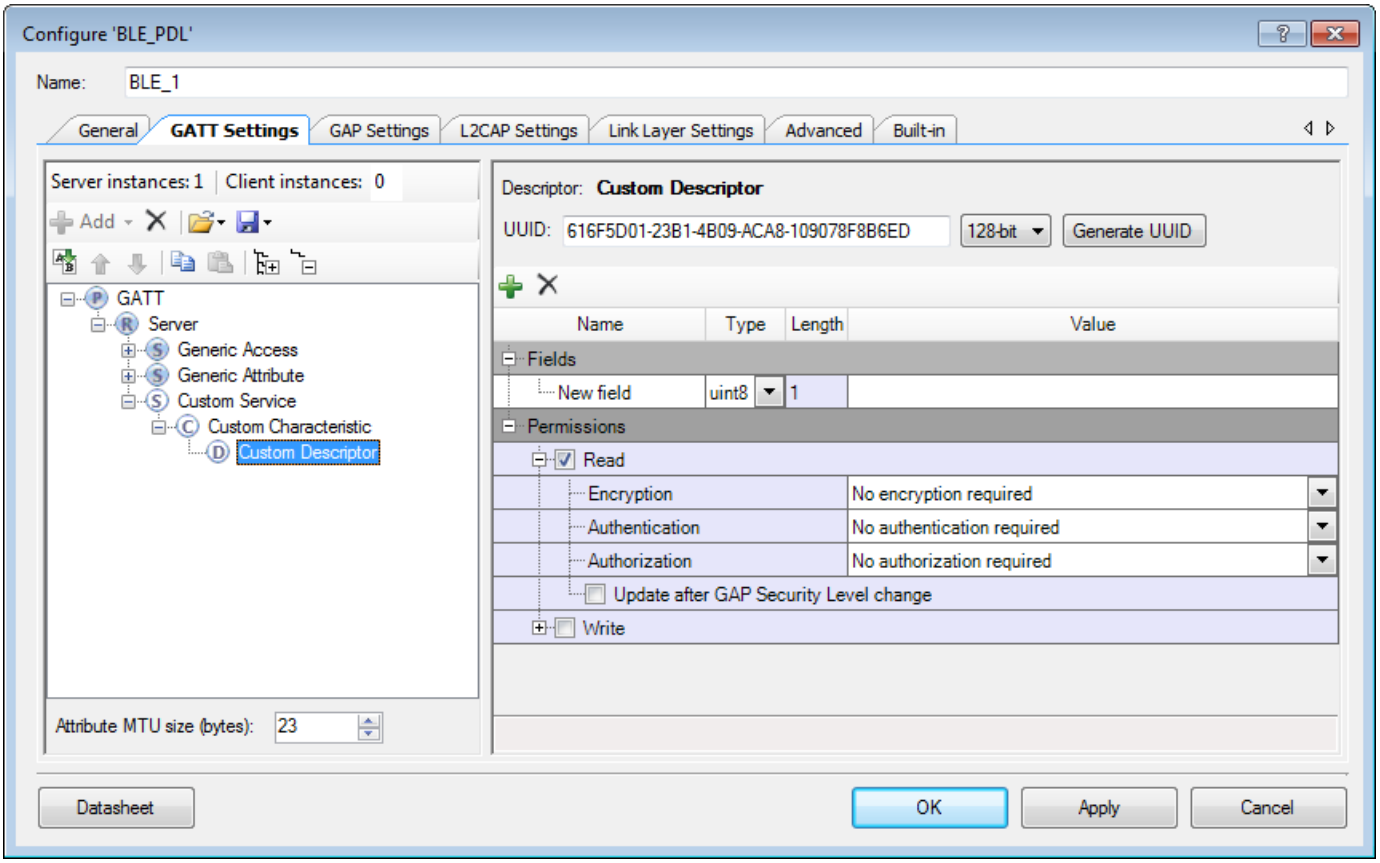
Properties（属性）

特性属性定义了特性值的使用方法。某些属性（Broadcast, Notify, Indicate, Reliable Write, Writable Auxiliaries）要求具有相应的特性描述符。有关详细信息，请参阅蓝牙内核规范第 3 卷、G 部分（GATT）、第 3.3.1.1 节“特性属性”的内容。

Permissions（权限）

特性权限定义了如何访问特性值属性以及进行访问所需要的安全级别。访问权限是根据特性属性设置的。在 **GAP Settings** 选项卡上安全配置 0 中更改了**安全模式**或**安全级别**参数时，**更改 GAP 安全级别后更新**复选框会确定是否自动更新了安全权限。其它安全配置不会影响属性权限。

自定义描述符配置



UUID

描述符的唯一通用标识符。该字段是自定义描述符中可编辑的字段。使用 **Generate** 按键随机生成 128 位 UUID。

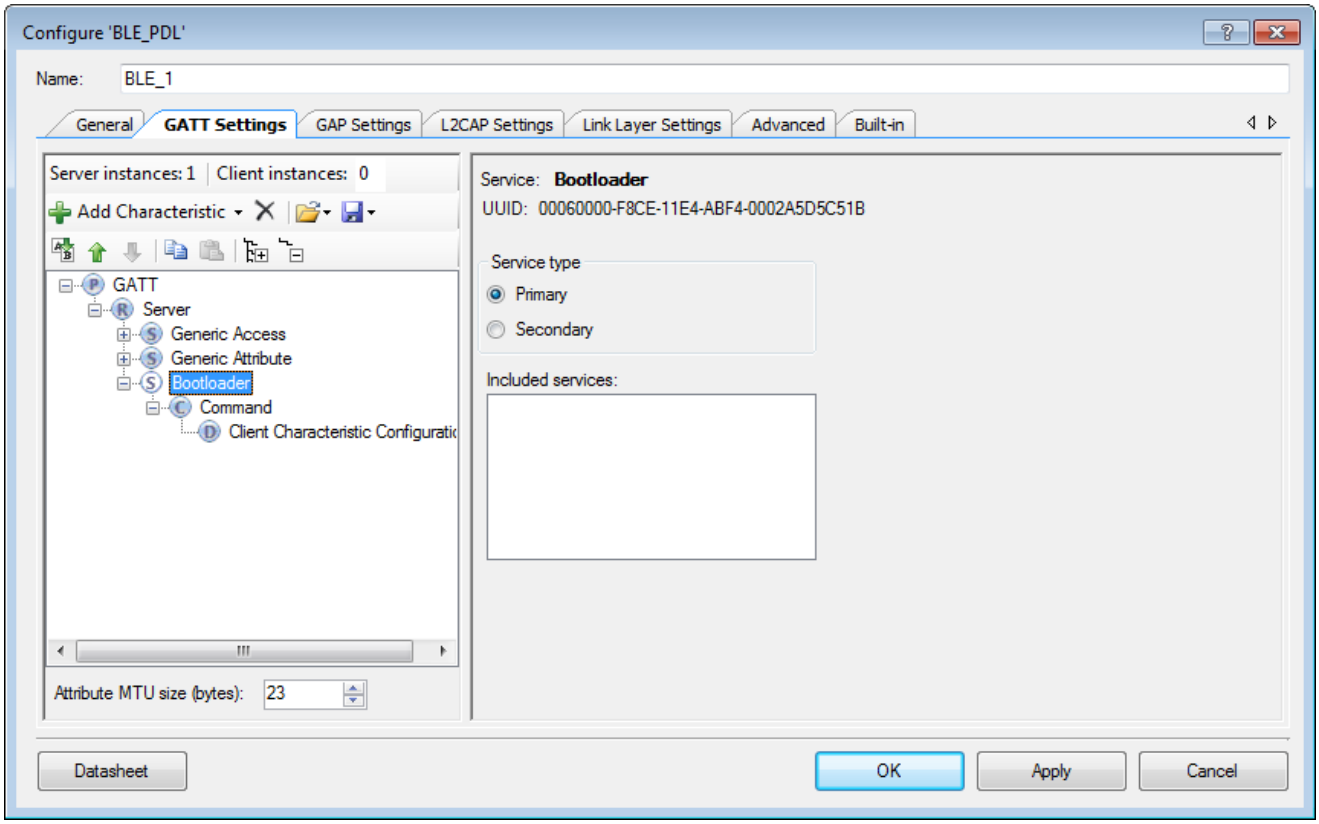
Fields（字段）

字段代表了描述符的值。可以在 **Value**（值）列中设置每个字段的默认值。在自定义描述符的情况下，可以自定义该字段。

Permissions（权限）

描述符权限定义了如何访问描述符属性以及该访问所需要的安全级别。

Bootloader 服务配置



UUID

服务的通用惟一标识符。UUID 被设置为 00060000-F8CE-11E4-ABF4-0002A5D5C51B。

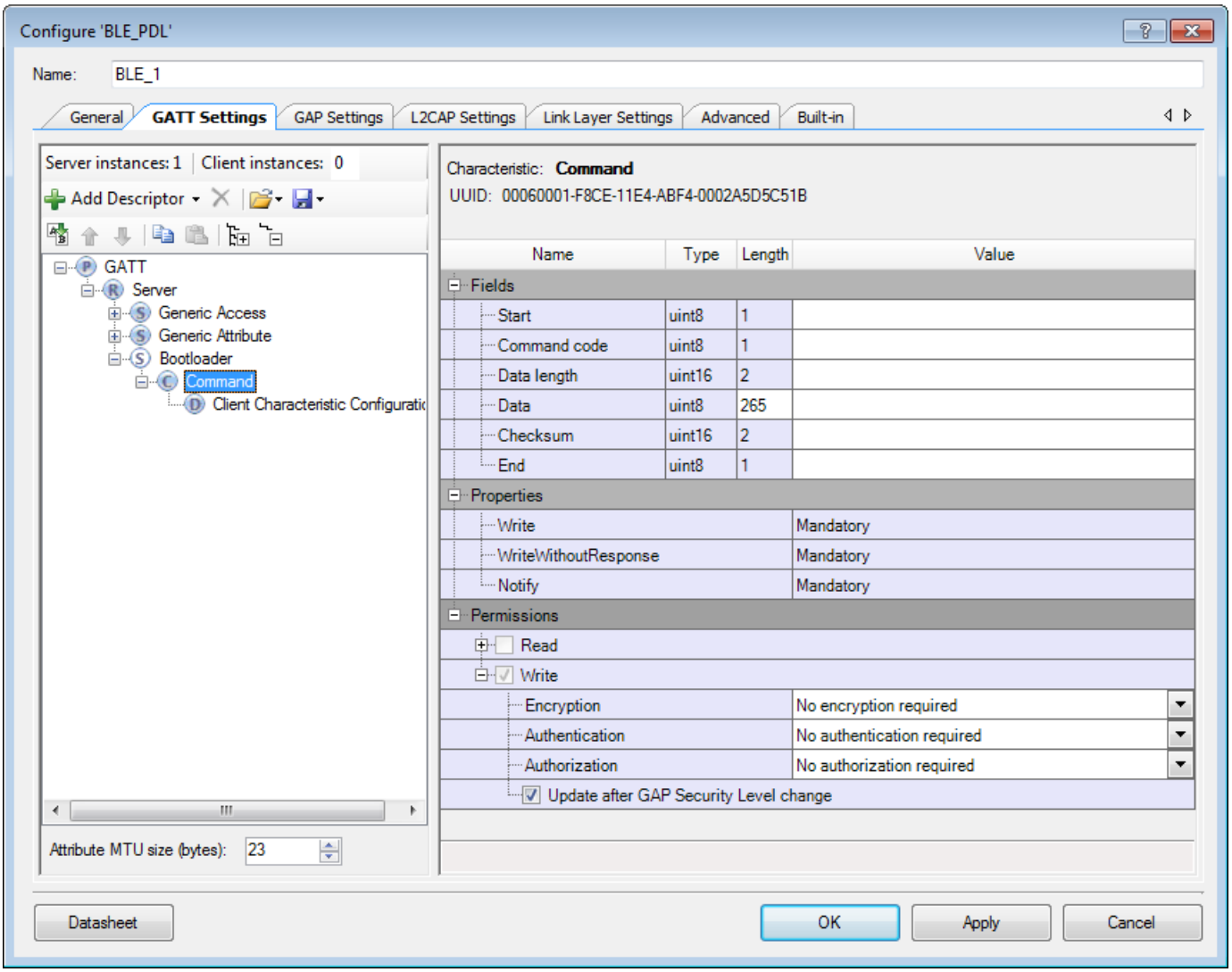
服务类型

- **Primary** — 表示器件的基本功能。
- **Secondary** — 表示器件的辅助功能。器件的辅助功能必须包含在另一个服务内。

所包含的服务

- 下表显示的是已选服务内可以包含的服务。每种服务能够包含一个或多个服务。所包含的服务为该服务提供辅助功能。

命令特性配置



UUID

特性的惟一通用标识符。UUID 被设置为 00060001-F8CE-11E4-ABF4-0002A5D5C51B。

Fields（字段）

字段代表命令特征值，如下所示。

- Start of packet（数据包开始） — 该常量定义了 Bootloader 数据包开始。
- Command（命令）— 该字段定义 Bootloader 命令。由于 Bootloader 命令依赖于赛普拉斯 Bootloader/Bootloadable 组件的修订版，因此请参阅 Bootloader/Bootloadable 组件数据表，以获取 Bootloader 命令的列表和描述。
- Status Code（状态代码） — 该字段定义了命令的状态代码。
- Data Length（数据长度）— 该字段定义了 Bootloader 命令/响应的长度。应该将该字段设置为设计中可使用的最大命令数据长度。应从 Bootloader 组件数据表中获取最大命令数据长度。

  根据 BLE 协议的细节，如果命令需要一个大于 20 字节的响应，则应增加属性 MTU 大小。要想支持数据长度被设置为 56 的响应（**Get Metadata** 命令响应），属性 MTU 大小应设置为 66。这可以从以下等式中看出：

  MTU 大小 = 数据长度 + Bootloader 命令开销 + 通知参数开销

  其中：

  - *数据长度* = 响应数据长度
  - *Bootloader 命令开销* = 7
  - *通知参数开销* = 3

  不符合该操作要求将导致 BLE_PDL 组件无法向所请求的命令发送响应。
- Data（数据）— 该字段定义 Bootloader 命令数据。该字段的长度由 Data Length 字段指定。
- Checksum（校验和） — 该字段定义针对完整数据包（但 Checksum 和 End of Packet 字段除外）进行计算的校验和。
- End of packet（数据包结束） — 该常量定义 Bootloader 数据包的结束。

Properties（属性）

可以写入或通知命令特性。

Permissions（权限）

特性权限定义了如何访问特性值属性以及进行访问所需要的安全级别。访问权限是根据特性属性设置的。**更改 GAP 安全级别后更新**复选框决定了在 GAP Settings 选项卡中更改**安全模式**或**安全级别**参数时是否自动更新安全权限。

GAP Settings（GAP 设置）选项卡

GAP 参数定义了连接蓝牙器件时所需要的通用连接设置。根据您在文件树中选择的项，它包含了大部分的这些参数。

根据 **General** 选项卡中所选定的 GAP 模式，**GAP Settings** 选项卡将显示可用的设置内容。通过这个选项卡可以使用 **Restore Defaults**（恢复默认设置）按键恢复默认的设置内容。

根据您在该文件树中选择的项，下述各部分会展示参数的不同目录。

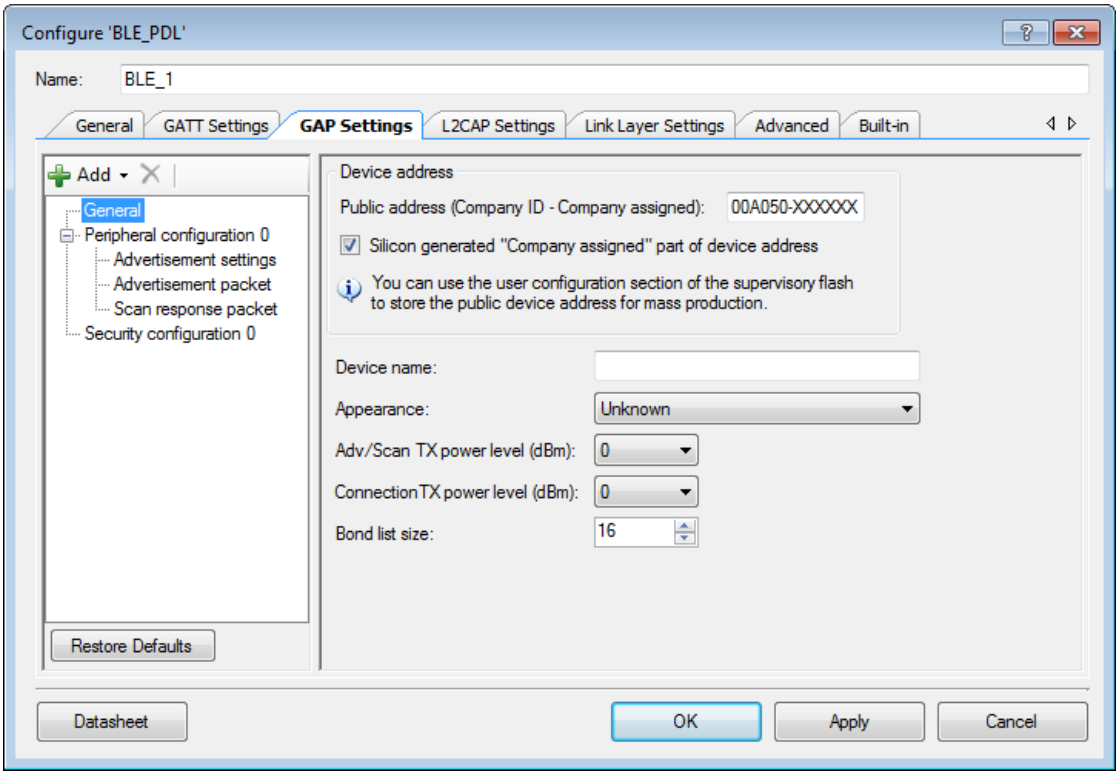
工具栏

工具栏提供了添加或删除 GAP 模式配置和安全配置的方法。

- **Add**（添加） — 允许添加外设、中心设备、广播器、观察器或安全配置。可用选项取决于 **General** 选项卡中选择的 GAP 模式。您可以为某个 GAP 模式添加多个配置，并使用固件在它们之间切换。
- **Delete**（删除） — 允许删除所选配置。

GAP Settings 选项卡–General

该部分包括通用 GAP 参数：



Public device address (Company ID – Company assigned)（公共器件地址（公司 ID — 公司分配））

该参数是用于识别器件的唯一一个 48 位蓝牙公共地址。它包括下面两个部分：

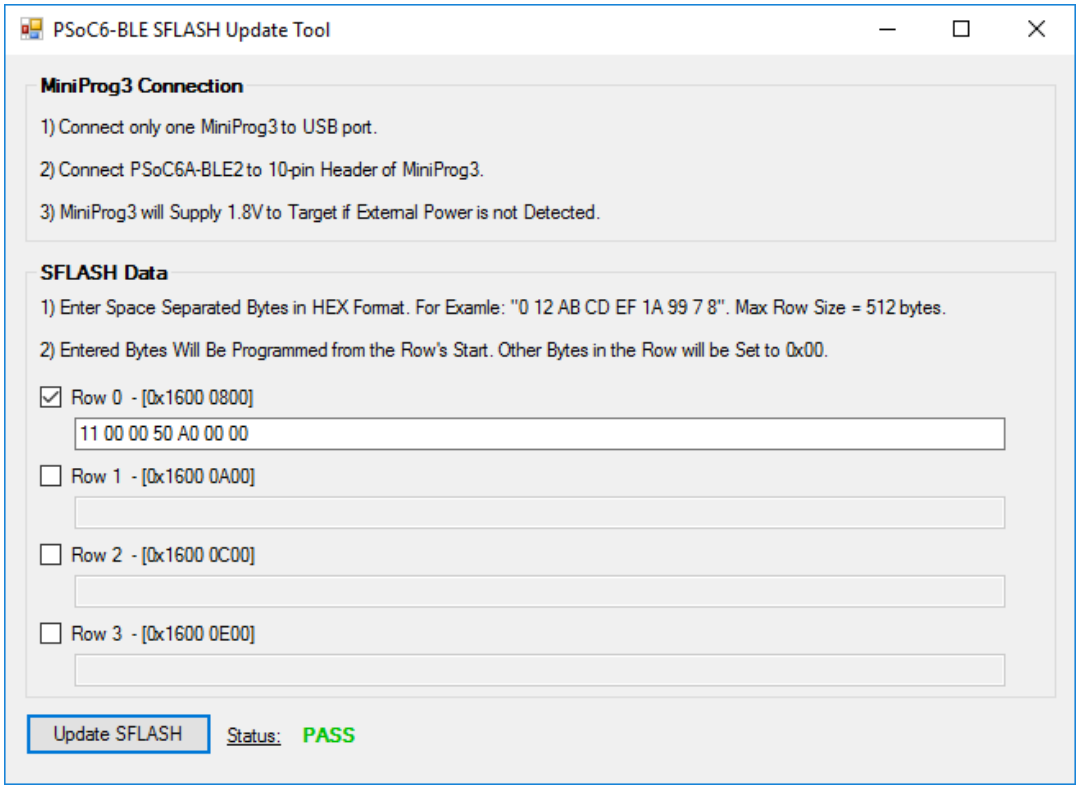
- **Company ID** 部分由 24 个最高有效位表示。该部分是由 IEEE 分配的 24 位组织唯一一个标识符（OUI）地址。
- **Company assigned** 部分由 24 个最低有效位表示。

该处所配置的地址是静态的，并且专门设计用于开发目的。在生产过程中，应通过 SWD 接口将器件地址编程到器件地址（用户 SFLASH 的第 0 行）的用户 SFLASH 位置上。一般情况下，在批量生产过程中，必须对该地址进行一次编程，这样在运行环境下它便不再发生任何变化。不过，在运行环境下可以多次对用户闪存进行配置。

在原型设计（固件设计）期间，可以使用 MiniProg3 和安装在以下 PSoC Programmer 文件夹中的示例应用将器件地址编程到用户的 SFLASH 位置：

C:\Program Files (x86)\Cypress\Programmer\Examples\Misc\PSoC6-BLE2-SFLASH-Update\Executable

在第 0 行中输入 cy_stc_ble_gap_bd_addr_t 类型的器件地址结构，并将其存储在 SFLASH 中。



组件不使用第 1 行、第 2 行和第 3 行，这些行用于存储用户信息。

在源代码中提供了该应用，并且它能够作为生产编程器中实现的参考示例。

Silicon generated “Company assigned” part of device address（器件地址中芯片所生成的“公司分配”部分）

选中该选框时，使用工厂编程的芯片 X/Y 位置、硅的晶圆 ID 和批次 ID 来生成器件地址中“公司分配”的部分。

Device Name（器件名称）

显示在对端端的器件名称。它默认拥有相应的读（未经验证/授权）和写属性。该参数最多可达 248 个字节。

注意：该参数用于配置 **Profile Tree**（配置文件树）中的 **GAP Service Device name**（GAP 服务器件名称）特性。当器件是一个 GATT 服务器时，可以修改该参数。

Appearance（外观）

显示器件的徽标或外观（它是 SIG 定义的 2 字节值）。它默认拥有相应的读（未经验证/授权）和写属性。

注意：该参数用于配置 **Profile Tree** 中的 **GAP Service Appearance**（GAP 服务外观）特性。当器件为 GATT 服务器时，可以修改该参数。

Adv/Scan TX power level（广播/扫描 TX 功耗级别）

启动时，广播或扫描通道的发送器的初始功耗级别（dBm）。默认值为：0 dBm。可选值为：-20 dBm、-16 dBm、-12 dBm、-6 dBm、0 dBm、4 dBm。

Connection TX power level（连接 TX 功耗级别）

启动时，连接通道的发送器的初始功耗级别（dBm）。默认值为：0 dBm。可选值为：-20 dBm、-16 dBm、-12 dBm、-6 dBm、0 dBm、4 dBm。

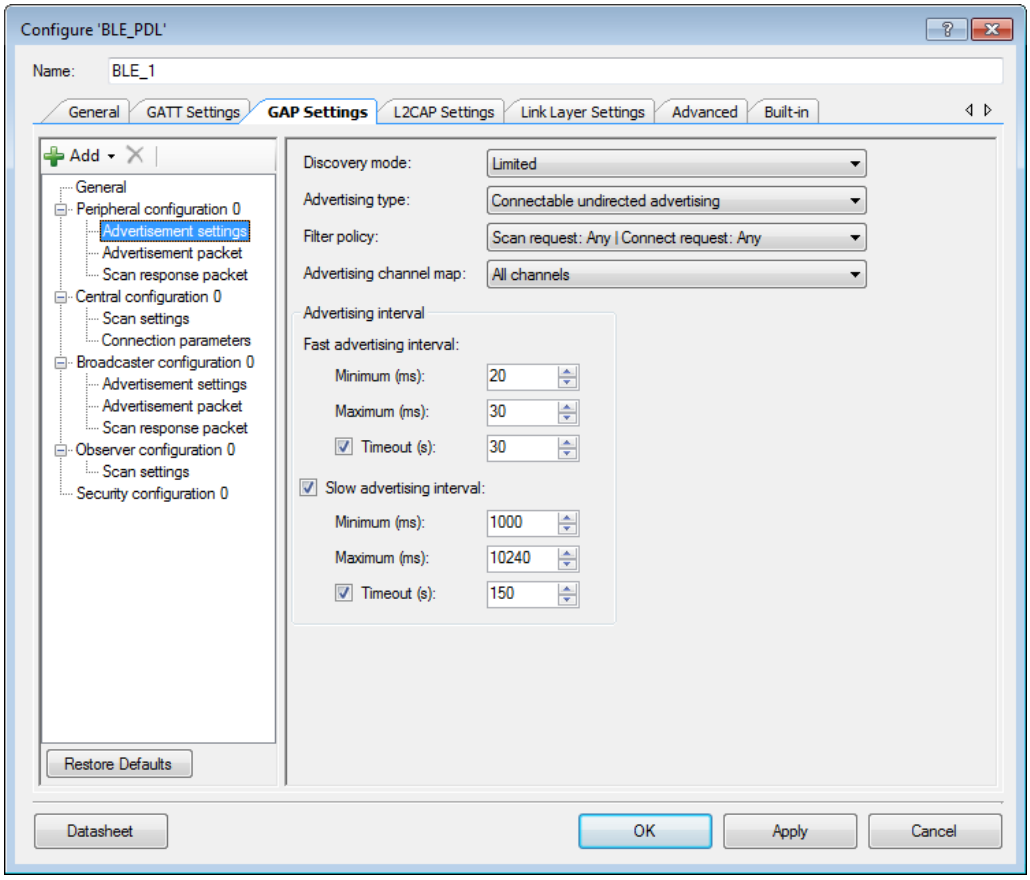
绑定列表大小

器件支持的最大绑定器件数。有效范围为 1 到 128。默认值：16

注意：绑定器件的最大数量也受限于存储数据所用的可用 Flash（模拟 EEPROM 区域）大小。受支持的服务数量和支持的绑定器件数量的倍数被计算为所消耗的 Flash 大小。

GAP Settings 选项卡 —–Advertisement Settings（广播设置）

当将器件配置为 Peripheral 或 Broadcaster 等 **GAP 模式**时，这些参数会可用的。



Discovery mode（检测模式）

- **Non-discoverable**（不可检测） — 在该模式下，中心器件不会检测到该器件。
- **Limited Discoverable**（受限检测模式） — 如果要求在有限的时间内（在暂时条件或特殊事件期间）检测到器件，请使用该模式。可以将受限检测模式中被广播的器件和执行受限检测规程的中心器件连接在一起。超时时长是由可应用的广播超时参数定义的。

- **General Discoverable**（通用检测） — 在该模式下，该器件必须由需要连续检测或需要进行无特殊条件检测的器件使用。配置为通用检测广播模式的器件可以被正在执行通用检测规程的中心器件所连接。

Advertising type（广播类型）

该参数定义了 LL 在适当的**检测模式**下使用的广播类型。

- **Connectable undirected advertising**（可连接的非定向广播） — 该选项用于对广播数据和扫描响应数据进行常规广播。它允许其它器件连接到该器件上。
- **Scannable undirected advertising**（可扫描的非定向广播） — 该选项用于将广播数据和扫描响应数据传播到有效的扫描器内。
- **Non-connectable undirected advertising**（无连接的非定向广播） — 该选项仅用于传输广播数据。

Filter policy（过滤政策）

该参数定义了如何对扫描和连接请求进行过滤。

- **Scan request: Any | Connect request: Any**（扫描请求：任意 | 连接请求：任意） — 程序扫描并连接所有器件中的请求。
- **Scan request: White List | Connect request: Any**（扫描请求：白名单 | 连接请求：任意） — 程序仅扫描白名单中的各器件，并连接所有器件中的请求。
- **Scan request: Any | Connect request: White List**（扫描请求：任意 | 连接请求：白名单） — 程序扫描所有器件中的请求，并连接白名单中各器件的请求。
- **Scan request: White List | Connect request: White List**（扫描请求：白名单 | 连接请求：白名单） — 程序扫描并连接白名单中各器件的请求。

Advertising channel map（广播通道映射）

该参数用于使能特定的广播通道。

- **Channel 37** — 使能第 37 号广播通道
- **Channel 38** — 使能第 38 号广播通道
- **Channel 39** — 使能第 39 号广播通道
- **Channel 37 and 38** — 使能第 37 和 38 号广播通道

- **Channel 37 and 39** — 使能第 37 和 39 号广播通道
- **Channel 38 and 39** — 使能第 38 和 39 号广播通道
- **All channels** — 使能全部三个广播通道

Advertising Interval（广播间隔）

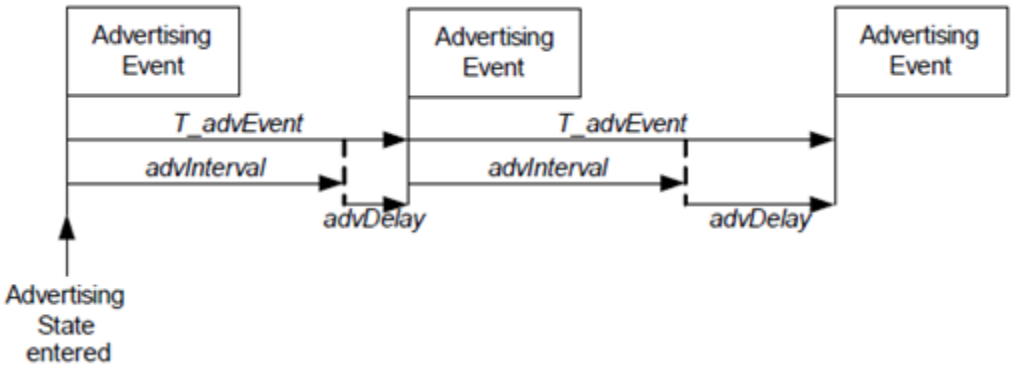
该参数定义了两个广播事件之间的间隔。设置两种广播间隔类型（即为**快速广播间隔**和**慢速广播间隔**）的最小值和最大值。初始化器件后，外设器件通常使用快速广播间隔。**Fast advertising interval timeout**（快速广播间隔超时）值到期后，如果还没有建立与中心器件的连接，那么配置文件会切换到慢速广播间隔，这样可以延长电池的使用寿命。**Slow advertising interval timeout**（慢速广播间隔超时）参数值到期后，将生成 CY_BLE_EVT_GAPP_ADVERTISEMENT_START_STOP。

注意：广播间隔要与选定的配置文件规范相对应。

注意：在 **General Discovery mode** 下，不支持超时。

- **Fast advertising interval**（快速广播间隔） — 该广播间隔允许更快的 LE 连接。当连接时间值位于已指定的最小值和最大值之间的范围内时，BLE_PDL 组件会使用该间隔值。
  - 最小值：是指广播数据和建立 LE 连接间所需的最小间隔。该参数的递增步长为 0.625 ms。其有效范围为 20 ms ~ 10240 ms。
  - 最大值：是指广播数据和建立 LE 连接之间所需的最大间隔。该参数的递增步长为 0.625 ms。其有效范围为 20 ms ~ 10240 ms。
  - 超时：是指通过快速广播间隔参数定义的广播超时值。取消选择后，器件会持续广播，并且慢速广播设置将变得不可用。在广播时间间隔结束前不能发生超时，所以如果超时值小于快速广播间隔的最小值，则会显示一次警告。该参数不应用于 **General discovery mode**。
- **Slow advertising interval**（慢速广播间隔） — 它定义了慢速广播间的时间间隔。它是一个可选参数。如果它被使能，那么会以更低的占空比实现广播，这样可延长电池的寿命。在内部快速广播间隔超时发生后，器件将应用 Slow advertising interval 参数。使用该参数所定义的最小和最大值允许 BLE 协议栈预估在这些间隔间发生的广播间隔连接。该参数不应用于 **General discovery mode**。
  - 最小值：是指广播数据和建立 LE 连接间所需的最小间隔。该参数的递增步长为 0.625 ms。有效范围为 1000 ms 到 10240 ms。
  - 最大值：是指广播数据和建立 LE 连接之间所需的最大间隔。该参数的递增步长为 0.625 ms。有效范围为 1000 ms 到 10240 ms。

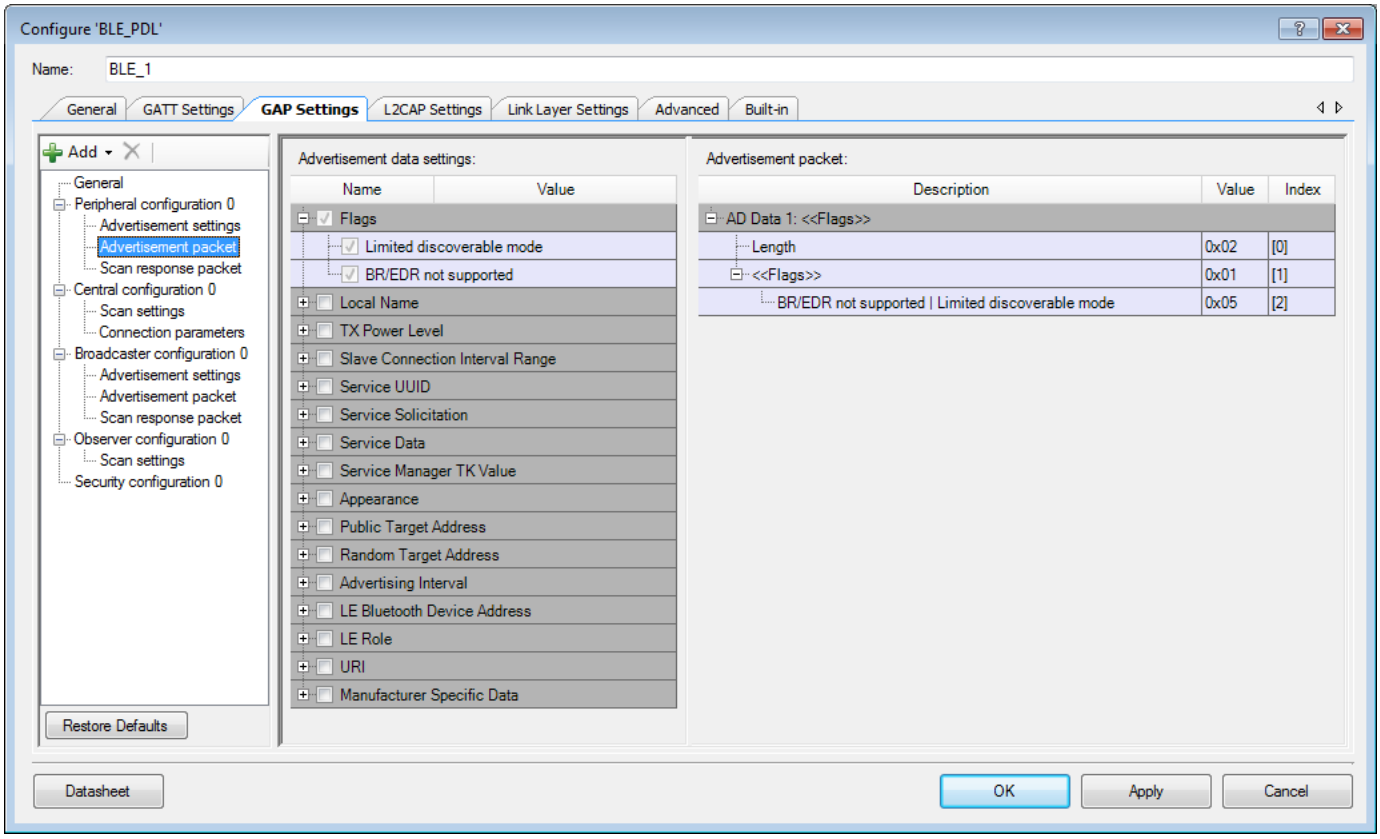
- 超时：通过慢速广播间隔参数定义的广播超时值。取消选择时，器件将连续进行广播。在广播时间间隔到期之前不能发生超时，所以如果超时值小于慢速广播间隔的最小值，则会显示警告。



  - AdvDelay 是伪随机延迟，其有效范围为 0 到 10 ms。
  - 完整的广播事件包括了每个所使用的广播通道上发送一个广播 PDU。

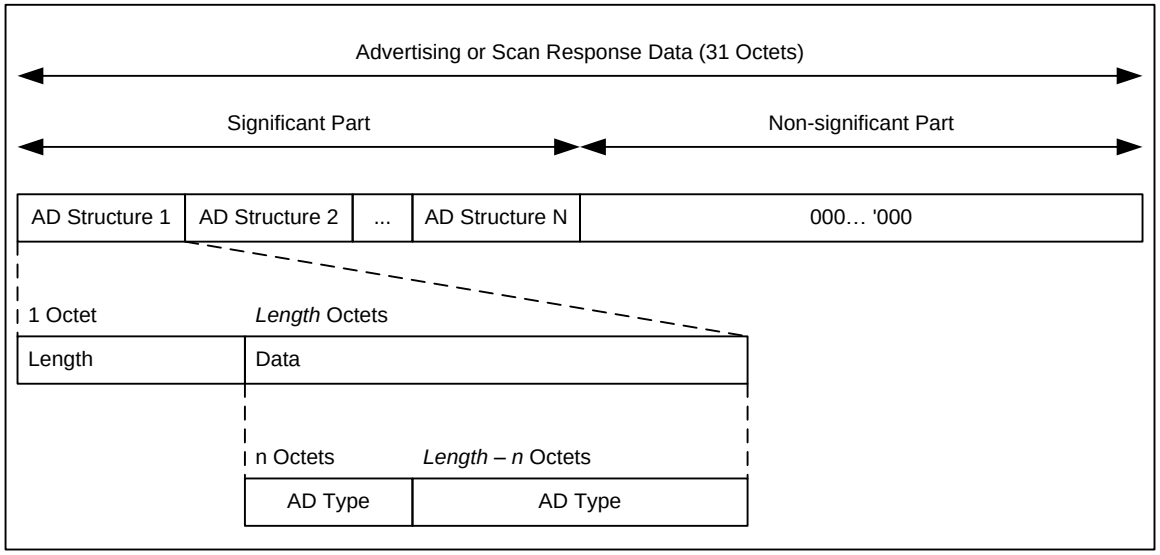
GAP Settings 选项卡 — Advertisement packet（广播数据包）

当将器件配置为支持外设或广播器等 **GAP 模式**时，将显示该部分。通过该项，可以配置 **Advertisement data**（广播数据），以便能够在器件广播中使用该数据。



广播/扫描响应数据设置

Advertisement (AD)（广播）或 **Scan response data**（扫描响应数据）数据包是一个大小为 31 字节的载荷，用于声明器件的 BLE 功能及其连接参数。下面介绍了这个数据的结构，如蓝牙 规范中所指定的内容。



该数据包可以包含多个广播结构。每个结构均由下面各个参数组成：

- **AD Length**（广播长度）：**AD Type**（广播类型）和 **AD Data**（广播数据）的大小，单位为字节。
- **AD Type**（广播类型）：广播结构中的广播类型。
- **AD Data**（广播数据）：与 **AD Type** 相关的数据。

整个广播数据包的总长度不能超过 31 个字节。

下面介绍了 **Advertisement data** 和 **Scan response data** 的示例结构。

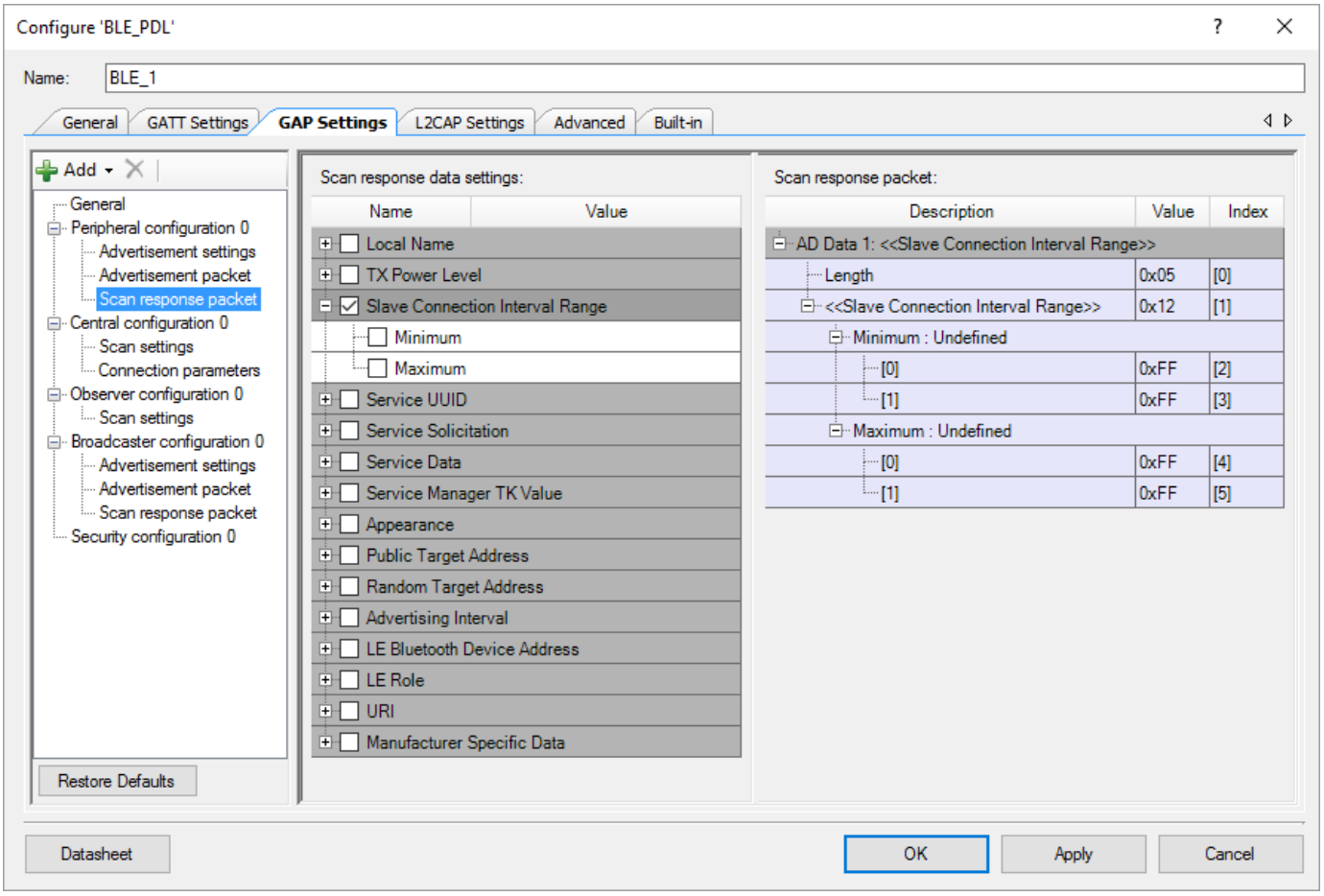
- 广播结构元素定义：
  - **AD Length**：**AD Type** 和相关 **AD Data** 的长度等于 5 个字节
  - **AD Type**（1 字节）：0x03 （服务 UUID）
  - **AD Data**（4 字节）：0x180D、0x180A（心率服务和器件信息服务）

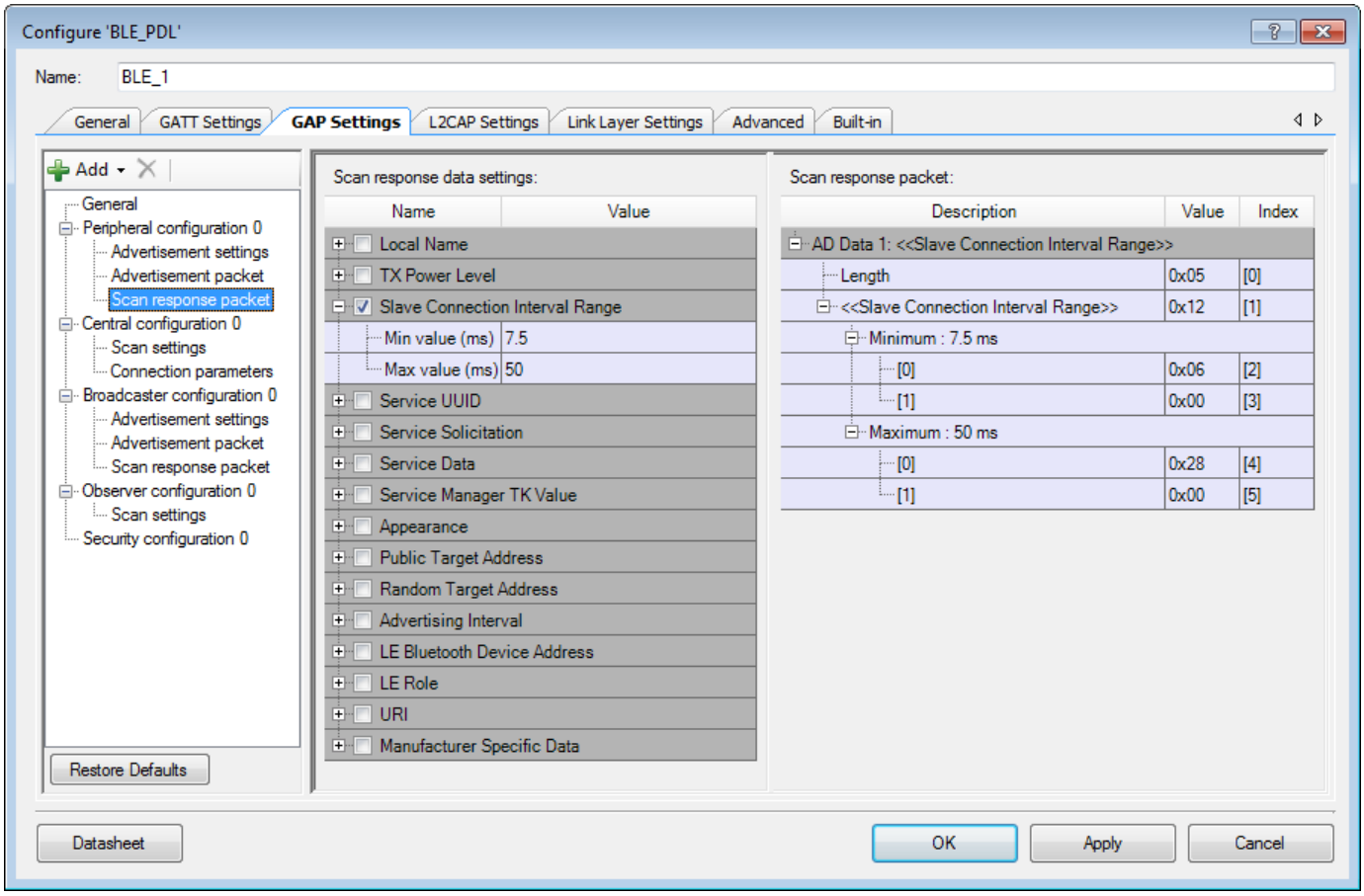
下表介绍了 **AD Types**（广播类型）。

广播类型	描述
标志	用于传播底层BLE传输功能的标志，如可检测模式、仅LE等。
本地名称	器件名称（完整的缩写名称）。该名称取自**GAP Settings**选项卡中**General**下面的**Device Name**字段。
Tx功耗级别	发送的功耗级别。它取自**GAP Settings**选项卡中**General**下的**Adv/Scan TX power level**字段。
从设备连接间隔范围	器件的首选连接间隔范围。不适用于**广播器**GAP模式。
服务UUID	器件所执行的用于传播的服务UUID列表。用于广播的AD类型值分别为16位、32位和128位服务UUID。如果16位和32位服务UUID是由Bluetooth SIG分配的，那么需要使用它们。
服务申请	外设将要使用中心器件的服务UUID列表。用于广播的AD类型值分别为16位、32位和128位服务UUID。
服务数据	2/4/16字节服务UUID，后面连接额外的服务数据。
安全管理TK值	在配对时使用的临时密钥。不适用于**广播器**GAP模式。
外观	器件外观。该值来自**GAP Settings**选项卡中**General**下的**Appearance**字段。
公共目标地址	指定目标的公共器件地址。
随机目标地址	指定目标的随机器件地址。
广播间隔	广播间隔值被计算为快速广播最大和最小间隔的平均值，这些值取自**GAP Settings**选项卡中**Advertisement Settings**字段的配置。
LE蓝牙器件地址	本地器件地址。该值来自**GAP Settings**选项卡上**General**下的**Public device address**字段。
LE模式	受支持的LE模式不适用于**广播器**GAP模式。
URI	URI被定义在IETF STD 66中。
制造商的特定数据	2字节的公司标识符后面是制造商的特定数据。
室内定位	室内定位服务规范中指定的数据。当配置文件中存在室内定位服务时，该功能可用。

GAP Settings 选项卡— Scan response packet（扫描响应数据包）

当将器件配置为支持外设或广播器等 **GAP 模式**时，将显示该部分。通过该部分可以配置扫描响应数据包，用以响应由 GATT 客户端器件执行的器件扫描。

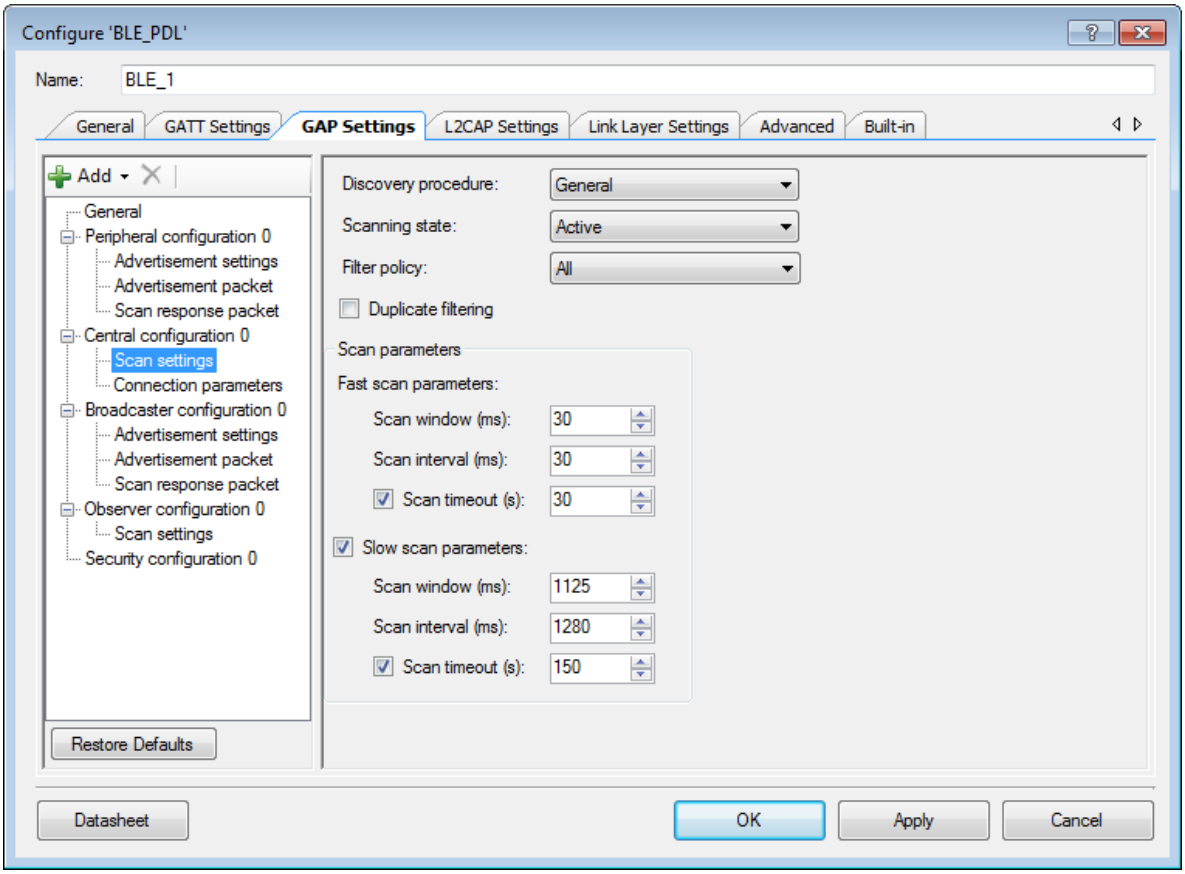




扫描响应数据包的结构与广播数据包的结构很相似。请参阅广播/扫描响应数据设置部分，了解有关配置扫描响应数据包的信息。

GAP Settings 选项卡 — Scan settings（扫面设置）

当将器件配置为中心器件或观察器等 **GAP 模式**时，这些参数会可用的。特别是在检测器件过程中，GATT 客户端器件将启动扫描规程。它在大约 30-60 秒的时间段内使用 **Fast scan parameters**（快速扫描参数），然后通过使用 **Slow scan parameters**（低速扫描参数）来降低扫描频率。



注意：扫描间隔要符合用户选定的配置文件规范。

Discovery Procedure（检测规程）

- **Limited**（受限制）— 执行该规程的器件会检测到只在受限发现模式下进行广播的器件。
- **General**（通用）— 执行该规程的期间会检测到在普通发现模式或首先发现模式下广播的器件。

Scanning state（扫描状态）

- **Passive**（被动） — 在该状态下，器件仅能监听广播数据包。
- **Active**（主动） — 在该状态下，器件会要求广播者提供更多的信息。

Filter policy（过滤政策）

该参数定义了如何过滤广播数据包。

- **All**（全部） **—** 处理所有广播数据包。
- **White List Only**（仅针对白名单） **—** 仅处理白名单中包含各器件的广播数据包。

Duplicate filtering（重复过滤）

如果被使能，它会激活重复对广播数据进行过滤的功能。如果被禁用，则 BLE 协议栈不会对广播数据执行过滤操作。

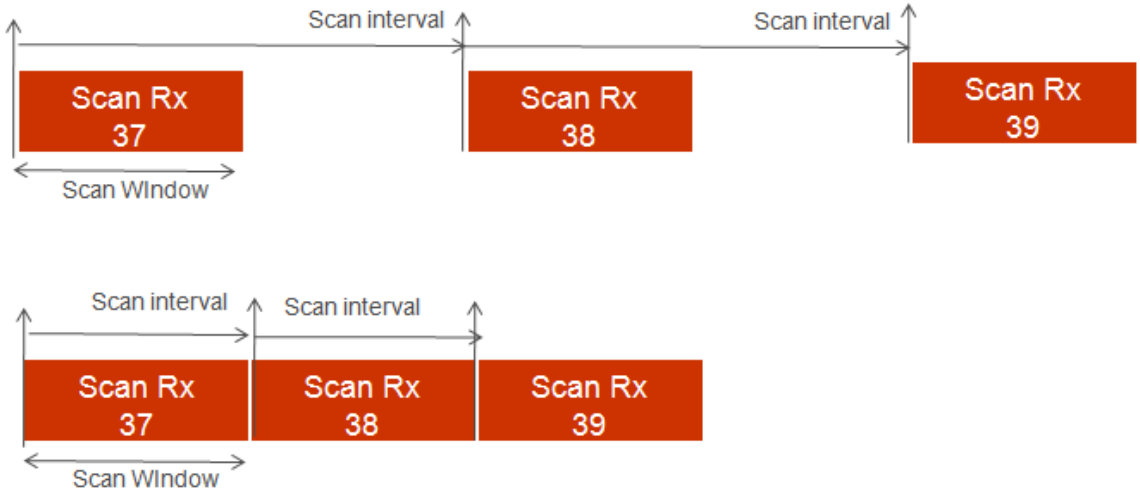
注意：控制器固件具有 8 个地址位置，用于缓存先前发现的广播器件，并对来自它们的重复数据包进行过滤。如果在扫描期间扫描仪附近存在超过 8 个广播器件，则地址存储缓冲区会被占完。用于覆盖地址高速缓冲存储器的固件算法以 FIFO 方式实现。当扫描程序发现超过 8 个广播器件时，在地址缓存中，第 9 个广播器件会替换掉第 1 个广播器件，第 10 个广播器件替换掉第 2 个广播器件，并依此类推。在地址缓存中替换掉第一个广播器件后，如果扫描程序再次看到第一个广播器件的 ADV 数据包，那么扫描程序会将其认为是一个新器件（因为第一个广播器件不被存储在地址缓存中），因此会将 ADV 数据包发送给主机。

Scan parameters（扫描参数）

这些参数用于定义扫描事件之间的扫描时间和间隔。使用了两个不同的扫描参数组：**Fast scan parameters** 和 **Slow scan parameters**。初始化器件后，中心器件通常使用快速扫描参数。**Fast scan timeout**（快速扫描超时）值到期后，如果还未建立与外设器件间的连接，那么配置文件会切换到慢速扫描间隔，这样可以延长电池的使用寿命。在 **Slow scan timeout** 值到期后，将生成 CY_BLE_EVT_GAPC_SCAN_START_STOP 事件。请参考 API 文档。

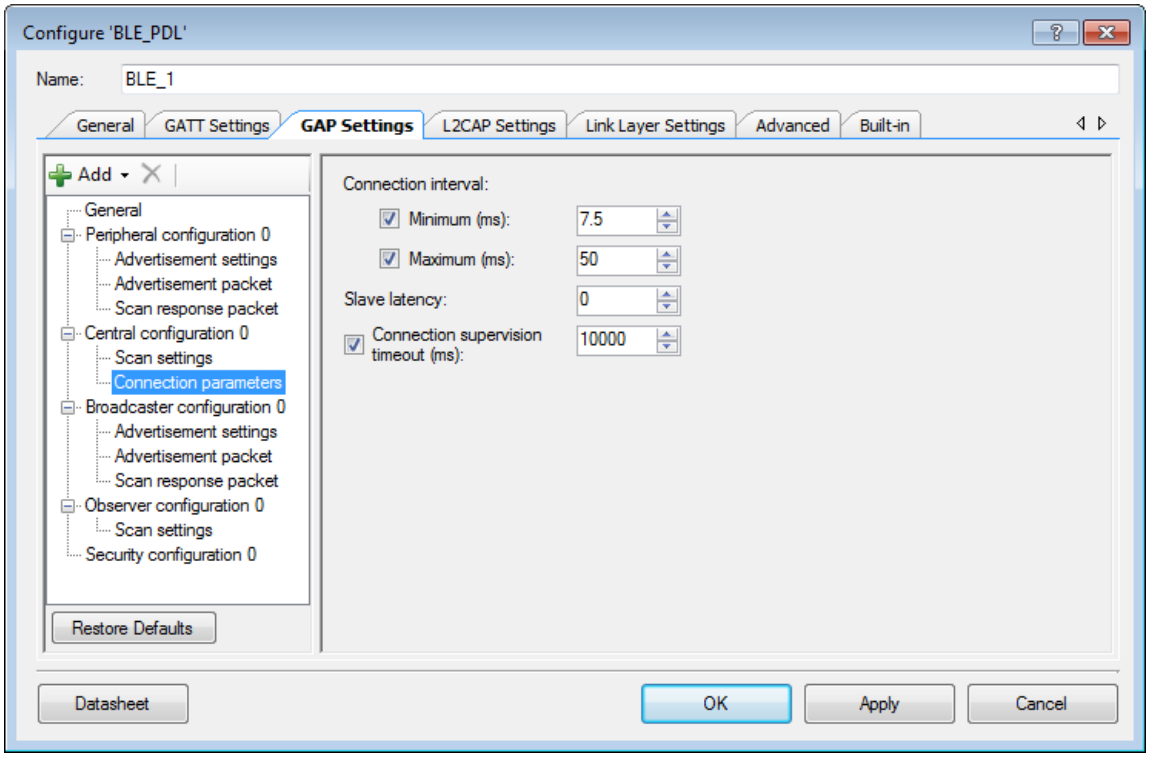
- **Fast scan parameters**（快速扫描参数）— 使用该类连接时，GATT 客户端和服务器之间的连接速度比普通连接的速度快。
  - **Scan Window**（扫描窗口）：当在 **Fast connection** 该参数的递增步长为 0.625 ms。其取值范围为 2.5 ms 到 10240 ms。**Scan Window** 值必须小于 **Scan Interval** 值。其默认值为 30 ms。
  - **Scan Interval**（扫描间隔）：当在 **Fast connection**（快速连接）该参数的递增步长为 0.625 ms。其取值范围为 2.5 ms 到 10240 ms。其默认值为 30 ms。
  - **Scan Timeout**（扫描超时）：使用快速扫描参数进行扫描时的超时值。默认值：30 s。取消选择时，器件将连续进行扫描。在扫描时间间隔到期之前不能发生超时，所以如果超时值小于慢速扫描间隔的最小值，则会显示警告。

- **Slow scan parameters**（低速扫描参数）— 使用该类连接时，GATT 客户端与 GATT 服务器之间的连接速度比普通连接速度慢。但该方法的功耗较低。
  - **Scan Window**（扫描窗口）：当在 **Slow connection** 该参数的递增步长为 0.625。其取值范围为 2.5 ms 到 10240 ms。**Scan Window** 值必须小于 **Scan Interval** 值。其默认值为 1125 ms。
  - **Scan Interval**（扫描间隔）：当在 **Slow connection** 该参数的递增步长为 0.625 ms。其取值范围为 2.5 ms 到 10240 ms。其默认值为 1280 ms。
  - **Scan Timeout**（扫描超时）：使用慢速扫描参数进行扫描时的超时值。默认值：150 s。取消选择时，器件将连续进行扫描。在扫描时间间隔到期之前不能发生超时，所以如果超时值小于慢速扫描间隔的最小值，则会显示警告。



GAP Settings 选项卡 — Connection 参数

这些参数定义了中心器件的首选 BLE 接口连接设置。



注意：BLE 协议栈会使用这些参数的调整值。这些值是无线发送的实际值。

- **Connection interval**（连接间隔） — 连接至外设器件的中心器件需要定义发生连接事件的时间间隔。
  - 最小值（单位为 ms）：该参数是连接事件期间所使用的最小连接时间值。它的步长为 1.25 ms。其取值范围为 7.5 ms 到 4000 ms。未选择该选项表示没有特定的最小值。
  - 最大值（单位为 ms）：该参数是连接事件期间使用的最大连接时间值。它的步长为 1.25 ms。其取值范围为 7.5 ms ~ 4000 ms。未选择该选项表示没有特定的最大值。

  注意：在多连接场合中，建议每个连接的最小连接间隔大于 N *完成 BLE 连接事件（CE）所用的最大连接时间。

  Min_CI = N * 每个 CE 的平均连接时间

  每个 CE 的平均连接时间指的是完成一个 BLE Tx 和 Rx 传输所占用的时间。该时间取决于链路层数据长度扩展功能（DLE）和 BLE 数据速率（1 Mbps 或 2 Mbps）的配置。应用可以使用以下时序查找表中的 CE 值：

  1. 如果启用了 DLE 且数据速率为 1 Mbps，则平均时间= 6 ms

2. 如果启用了 DLE 且数据速率为 2 Mbps，则平均时间 = 3.5 ms
3. 如果禁用了 DLE 且数据速率为 1 Mbps，则平均时间 = 2 ms
4. 如果禁用了 DLE 且数据速率为 2 Mbps，则平均时间 = 1.6 ms

例如，如果应用支持 4 个启用 DLE 并且 1 Mbps 数据速率的 BLE 连接，则每个连接的建议最小连接间隔为：

4 * 6 = 24 ms。

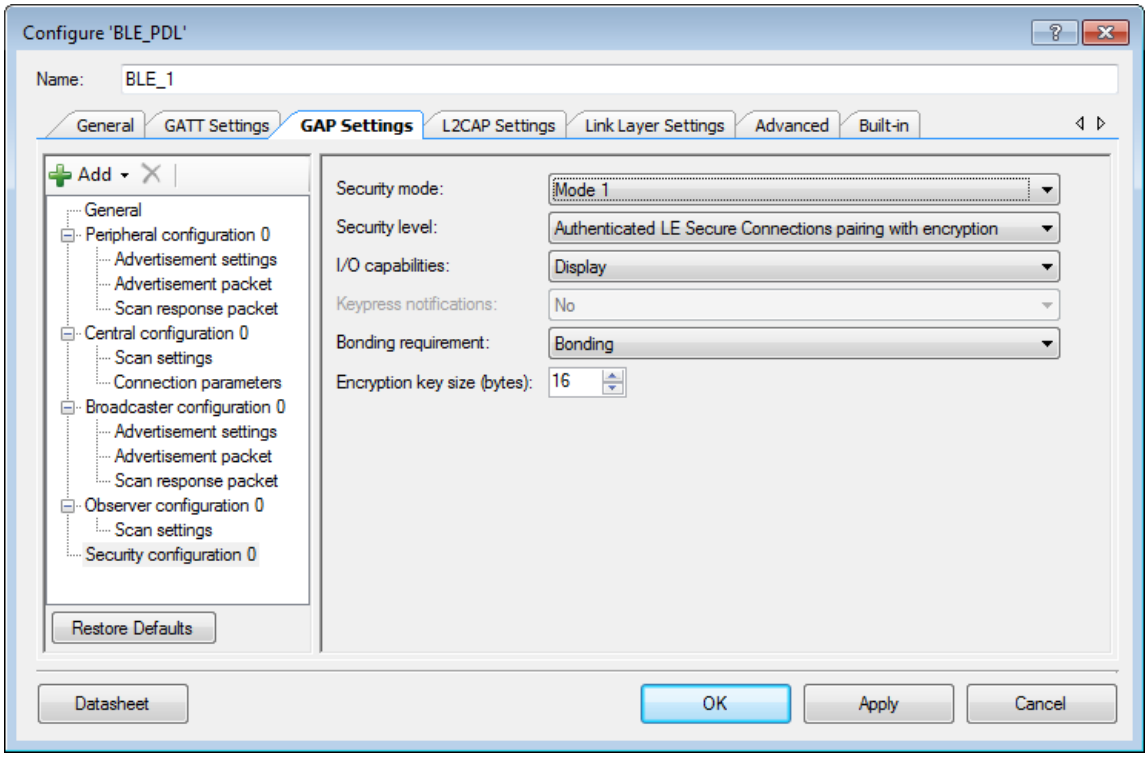
注意：小于该值的连接间隔仍然有效，但在某些情况下，具有较短间隔的更新实例的实时控制程序（连接更新、通道映射更新等）可能导致链路断开。

- **Slave Latency**（从设备延迟） — 定义了在连续连接事件中从设备响应一个连接事件时的延迟。它由多个连接间隔的术语表达，其中在每个间隔内只能发生一次连接事件。其取值范围为 0 到 499 个事件。
- **Connection Supervision Timeout**（连接监控超时）— 该参数定义了 LE 链接监控的超时间隔。在对端器件没有通过 LE 链接发出响应的情况下，该参数用于指定 LE 链接需要持续的超时时长。时间间隔被配置为 10 ms 的倍数。未选择该选项表示没有特定值。其取值范围为 100 ms 到 32000 ms。

  请注意，为了正常运行，连接监控超时必须大于**（1 +从设备延迟）*连接间隔* 2**（ms）。有关连接监控超时的更多信息，请参阅蓝牙内核规范第 6 卷、B 部分、第 4.5.2 章。

GAP Settings 选项卡 — Security（安全）

该部分包含了用于配置组件的全局安全选项的几个参数。只有选择了可连接的 GAP 模式（外设或中心器件）时，才能配置这些参数。可以通过使用 **Profile Tree** 中其独有的安全设置对每个特性进行相关设置。



Security mode（安全模式）

定义组件的 GAP 安全模式。两种可用的模式均支持验证操作。

- Mode 1（模式 1）— 用于要求数据加密的设计中。
- Mode 2（模式 2）— 用于要求数据签名的设计中。

Security level（安全级别）

根据所选定的 **Security mode**（安全模式）使能不同的安全级别：

- 如果选择了 Mode1，那么下面各安全级别可用。
  - No Security（无安全）— 如果应用该安全级别，器件将不会进行加密和验证操作。
  - Unauthenticated pairing with encryption（未经验证配对但实现加密操作）— 使用该安全级别时，在与远程器件建立连接后，器件将发送被加密的数据。

  - Authenticated pairing with encryption（经验证配对且实现加密）— 使用该安全级别时，在与远程器件建立连接后，器件将发送被加密的数据。要想建立连接，器件需要执行验证配对程序。
  - Authenticated LE Secure Connections pairing with encryption（经验证配对的 LE 安全连接与实现加密） — 通过这种安全级别，器件会使用一个称作椭圆曲线 Diffie-Hellman（ECDH）的算法进行密钥生成，并使用新的配对程序进行密钥交换。它还提供了一种新的中间人（MITM）攻击保护方法 — 即数字比较法。

- 如果选择了 Mode2，那么下面各安全级别可用。
  - Unauthenticated pairing with data signing（未经验证配对而实现数据签名）— 如果使用该安全级别，那么在发送数据前，器件将对其进行签名并与远程器件建立连接。
  - Authenticated pairing with data signing（经验证配对且实现数据签名）— 如果使用该安全级别，那么在发送数据前，器件将对其进行签名并与远程器件建立连接。要想建立所需连接，器件需要执行验证配对程序。

按键通知

该方法提供了在 LE 安全配对过程中的键盘器件选项，便于在用户输入或删除密钥时发送按键通知。当 **Security level** 被设置为“Authenticated LE Secure Connections pairing with encryption”并且 **I/O capabilities** 选项被设置为“Keyboard”或“Keyboard and Display”时，该选项可用。

I/O capabilities （I/O 功能）

该参数是指用于使能或限制特定配对方法或安全级别的器件输入和输出功能。

- Display （显示屏）— 用于带有显示功能的器件，能够显示验证数据。要求 GAP 验证。
- Display Yes/No（显示屏与是/否按键）— 用于带有显示屏并至少带有两个是/否输入按键的器件。要求 GAP 验证。
- Keyboard（键盘）— 用于带数字键盘的器件。要求 GAP 验证。
- No Input No Output（无输入无输出）— 用于不能输入或显示验证密钥数据的器件。用于鼠标类的器件。无需 GAP 验证。
- Keyboard and Display（键盘和显示屏）— 用于像 PC 和平板电脑类的器件。要求 GAP 验证。

绑定要求

该参数用于配置各项绑定要求。绑定的目的是为了在基于共同链接密钥（绑定）的两个蓝牙器件之间建立关系。在绑定过程中创建并交换（配对）链接密钥，并且预期两个蓝牙器件均存储该密钥，以供将来验证。可以绑定的最大远程器件数为 128。

- **Bonding**（绑定）：与远程器件配对后，该器件将在闪存存储器中保存连接的连接的链接密钥。如果该连接被断开并被重新建立，那么该器件将使用先前存储的密钥建立连接。

  注意：绑定信息被存储在 RAM 内，如果需要在关闭期间保留该信息，那么需要将它写入到闪存内。更多有关绑定和闪存写入的使用情况，请参考功能说明部分。

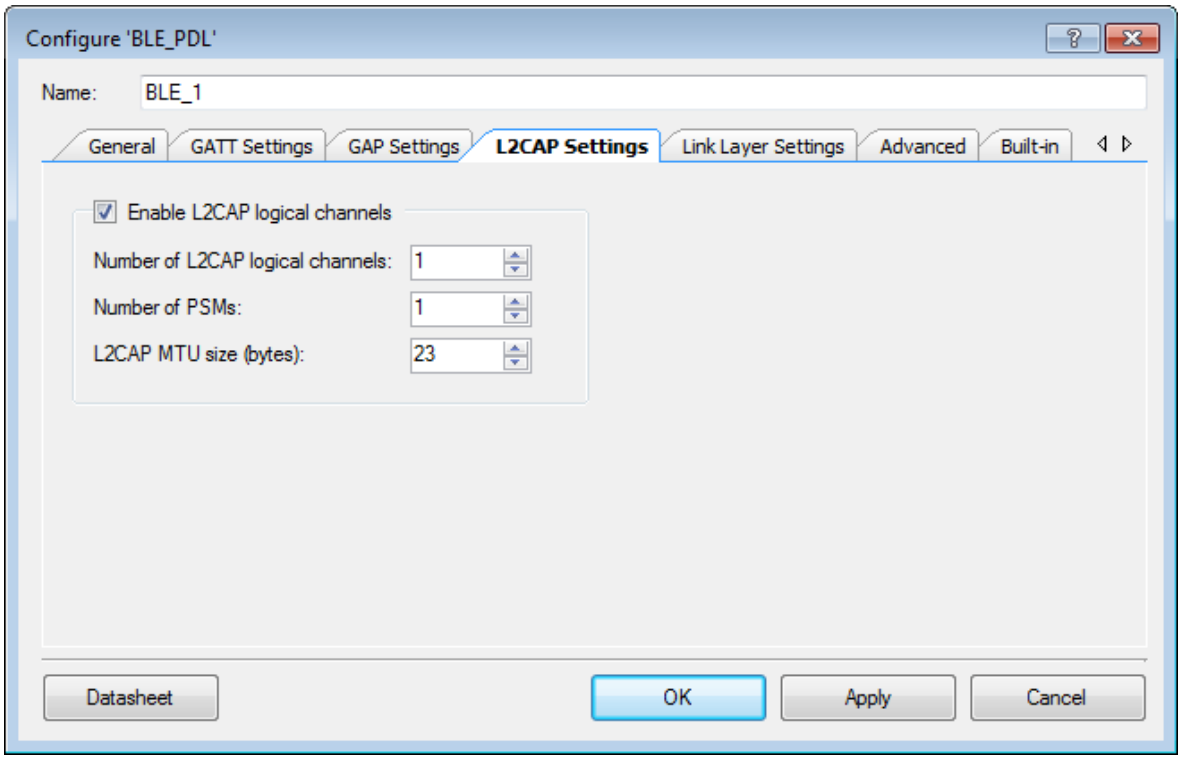
- **No Bonding**（无绑定）：每次建立连接时，将执行配对过程。

Encryption Key Size（加密密钥大小）

该参数会根据配置文件的要求定义加密密钥的大小。加密密钥大小的有效范围为 7 到 16 个字节。

L2CAP Settings 选项卡

L2CAP 设置定义了面向通道配置的 L2CAP 连接的参数。



Enable L2CAP logical channels（启用 L2CAP 逻辑通道）

该参数用于启用 L2CAP 逻辑通道的配置。默认值为：True。

Number of L2CAP logical channels（L2CAP 逻辑通道数量 ）

该参数用于定义应用所需面向逻辑通道的 LE L2CAP 连接数量。其取值范围为 1 到 255。默认值：1

Number of PSMs（PSM 数量）

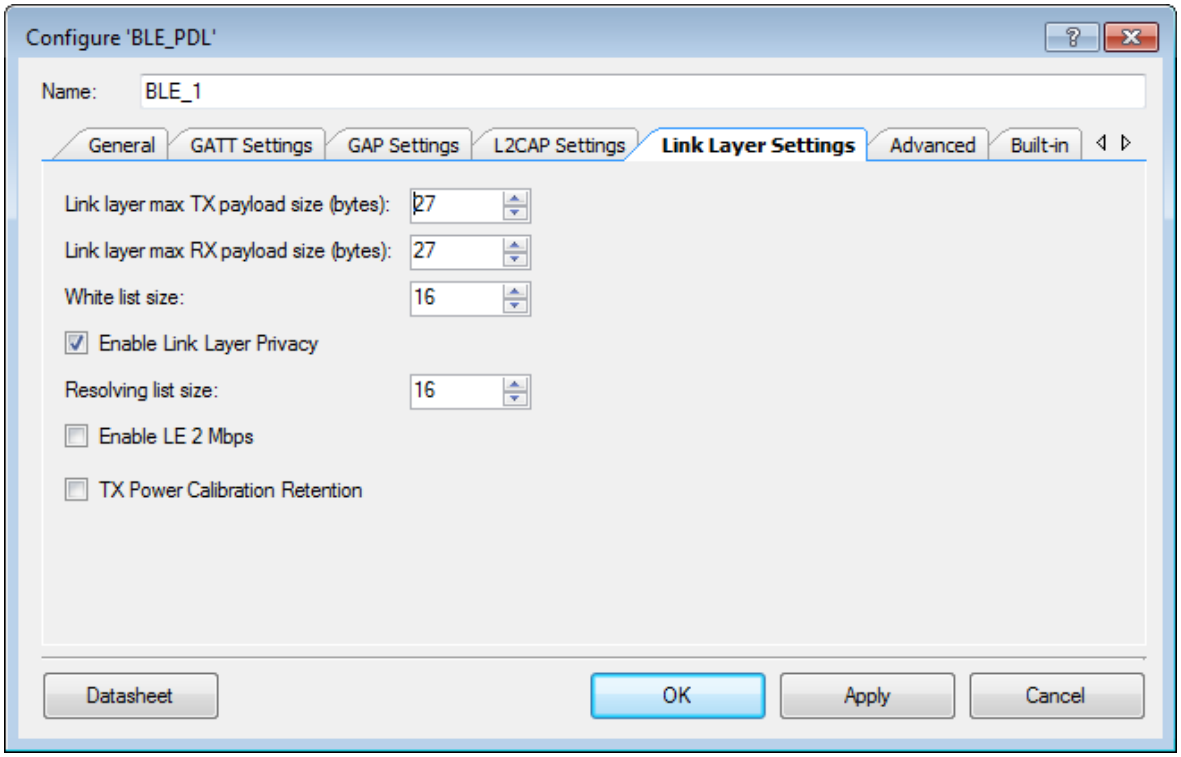
该参数用于定义应用所需的 PSM 数量。其取值范围为 1 到 255。默认值：1

L2CAP MTU size（L2CAP MTU 大小）

该参数用于定义 L2CAP 数据包的最大 SDU 长度。其取值范围为 23 到 65488 个字节。默认值：在支持**互联网协议支持服务**的情况下为 1280 个字节，否则为 23 个字节。

Link Layer Settings 选项卡

链路层设置参数。



Link Layer Max Tx Payload Size（链路层最大 Tx 有效载荷大小）

设计中会使用的最大链路层发送有效载荷大小。在数据长度更新过程中，根据对端器件的链路层接收数据包大小来确定链路层发送数据包的实际大小，并且通过 CY_BLE_EVT_GAP_DATA_LENGTH_CHANGE 事件来通知该实际大小。其取值范围为 27 到 251 个字节。

Link Layer Max Rx Payload Size（链路层最大 Rx 有效载荷大小）

设计中会使用的最大链路层接收有效载荷大小。在数据长度更新过程期间，基于对端器件的链路层发送数据包大小来确定链路层接收数据包的实际大小，并且将通过 CY_BLE_EVT_GAP_DATA_LENGTH_CHANGE 事件来通知该实际大小。其取值范围为 27 到 251 个字节。

将链路层最大 Tx 有效载荷大小或链路层最大 Rx 有效载荷大小设置为大于 27 的值，这样可启用 LE 数据长度扩展功能。

White list size（白名单大小）

可以添加到白名单中的最大器件数量。其取值范围为 1 到 16。默认值：16

Enable Link Layer Privacy（启用链路层保密）

启用蓝牙 4.2 的 LL Privacy 1.2 功能，并支持生成 CY_BLE_EVT_GAP_ENHANCE_CONN_COMPLETE 和 CY_BLE_EVT_GAPC_DIRECT_ADV_REPORT 事件。

请注意，启用该功能时不会生成 CY_BLE_EVT_GAP_DEVICE_CONNECTED 事件。

Resolving list size（解析表大小）

该器件应解析地址的最大对端器件数。启用 **Enable Link Layer Privacy** 功能时，该参数可用。其取值范围为 1 到 16。默认值：16

Enable LE 2 Mbps（启用 LE 2 Mbps）

通过该项可启用蓝牙 5.0 的 LE 2 Mbps 特性。

2 Mbps 特性支持新的物理（PHY）调制方案，从而可以增加两个支持该特性的器件间数据的吞吐量。更多有关该特性的详细信息，请参阅蓝牙核心规范版本 5.0。

发生 CY_BLE_EVT_STACK_ON 事件后，请使用 Cy_BLE_SetDefaultPhy() API 为所有连接设置默认首选的 PHY，或者使用 Cy_BLE_SetPhy() API 来设置当前连接的 PHY。

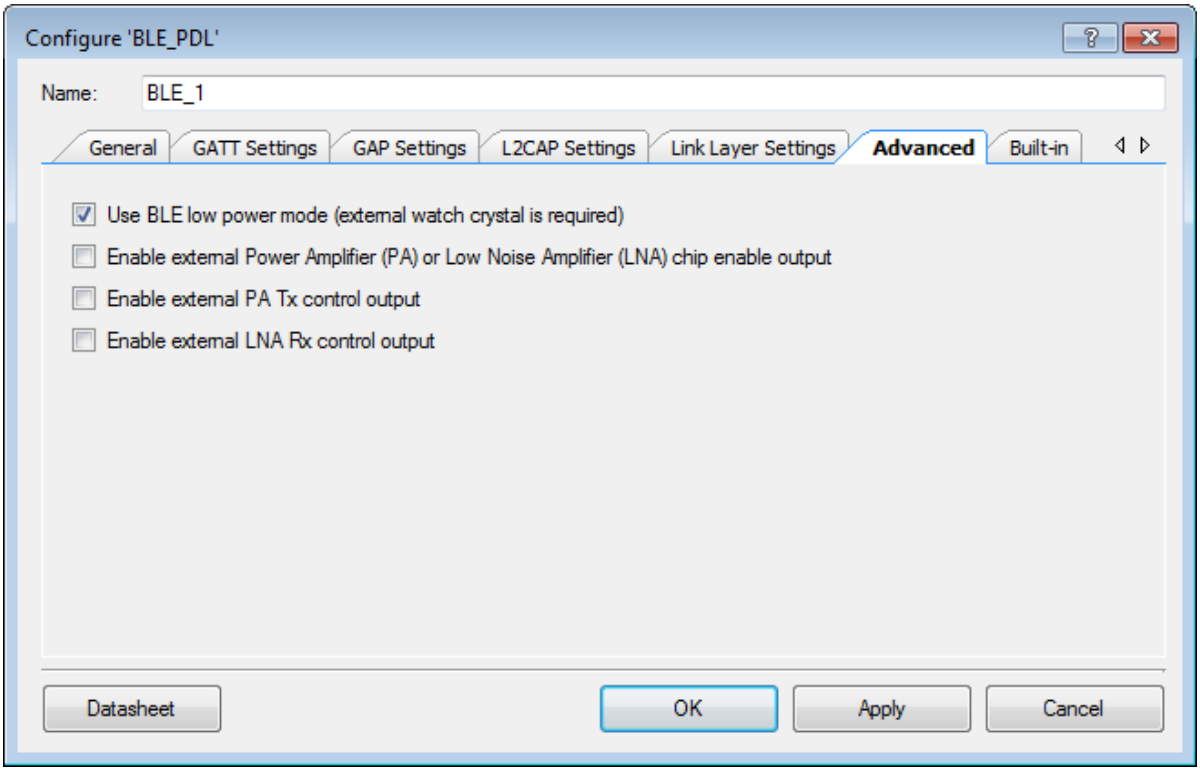
当控制器更改正在使用的发送器 PHY 或接收器 PHY 时，将指示 CY_BLE_EVT_PHY_UPDATE_COMPLETE 事件。

Enable Tx Power Calibration Retention（启用 Tx 功耗校准保留）

启用该特性后，在编程后仅执行一次 BLE 无线电 Tx 功率校准，并且校准值保留在 SFLASH 位置上。在连续设备重启过程中，该保留值被重新加载到无线电功率校准寄存器内。这样能够大大降低 BLE 协议栈启动时间

Advanced 选项卡

Advanced 参数定义了低功耗模式和外部功率放大的参数。



Use BLE low power mode（使用 BLE 低功耗模式）

通过该参数可以确定 BLE_PDL 组件是否需要支持低功耗模式。默认值为：True。

设置该参数时，必须选择 WCO 或 PILO 为设计范围资源时钟编辑器中的 LFCLK 源。如果您打算在低功耗模式下使用组件，则需要配置该参数。

Enable external Power Amplifier (PA) or Low NoiseAmplifier (LNA) chip enable output（使能外部功率放大器（PA）或低噪声放大器（LNA）芯片使能输出）

通过该参数可以启用外部 PA 和 LNA 芯片使能控制引脚。默认值为：False。

Enable external PA Tx control output（使能外部 PA Tx 控制输出）

选择该项后，来自 BLE 射频组件的 ext_pa_tx_ctl_out 信号将在 GPIO 上被路由。默认值为：False。

Enable external LNA Rx control output（使能外部 LNA Rx 控制输出）

选择后，来自 BLE 射频组件的 ext_lna_rx_ctl_out 信号将在 GPIO 上路由。默认值为：False。

默认情况下，所有 PA 输出均为高电平有效模式。发生 CY_BLE_EVT_STACK_ON 事件后请使用 Cy_BLE_ConfigureExtPA() API 来修改默认的输出极性。

应用编程接口（API）

应用编程接口（API）由 PDL 的 BLE 中间件库提供。成功编译后，中间件将被复制到应用项目的“pdl\middleware\ble\”目录中。

组件生成全局变量和预处理器宏部分所描述的配置结构。将生成的数据结构传递给应用初始化代码中的关联 BLE 中间件函数，用于配置外设。一旦初始化外设后，应用代码可以通过中间件 API 函数执行运行时更改。

BLE 中间件包含一个全面的 API 列表，这样您可以使用软件配置 BLE 协议栈、底层芯片硬件和 BLE 服务特定配置。您可以使用这些配置访问协议栈的 GAP、GATT 和 L2CAP 层。

有关全面 API 列表的详细说明，请参阅 BLE 中间件库文档。要访问该文档，请右键单击原理图上的 Component 符号，然后从下拉菜单中选择 **Open PDL Documentation…**选项。

除了 PDL API 之外，BLE_PDL 组件还提供了基于实例的组件 API，以及 PSoC Creator 提供的其它功能。

请注意，所有 BLE_PDL 组件 API 的名称均以 Cy_BLE_开始。这是 BLE_PDL 组件的特性，并且只允许将一个组件实例放入到您的设计中。

全局变量

BLE_PDL 组件提供了以下外设初始化数据结构。

uint8_t cy_ble_initVar = 0u

指示BLE是否已被初始化。该变量被初始化为0并在第一次调用Cy_BLE_Start()时设置为1。这样，第一次调用Cy_BLE_Start()子程序后，组件不用重新初始化即可重启。如果需要组件重新初始化，在调用Cy_BLE_Start()子程序前要将该变量设置为0。或可调用Cy_BLE_Init()函数重新初始化BLE。

cy_stc_ble_config_t cy_ble_config
BLE的配置结构。

cy_stc_ble_aios_config_t cy_ble_aiosConfig
自动化输入输出服务的配置结构。

cy_stc_ble_ancs_config_t cy_ble_ancsConfig
苹果通知中心服务的配置结构。

cy_stc_ble_ans_config_t cy_ble_ansConfig
警报通知服务的配置结构。

cy_stc_ble_bas_config_t cy_ble_basConfig
电池服务的配置结构。

cy_stc_ble_bcs_config_t cy_ble_bcsConfig
身体组成服务的配置结构。

cy_stc_ble_bls_config_t cy_ble_blsConfig
血压服务的配置结构。

cy_stc_ble_bms_config_t cy_ble_bmsConfig
绑定管理服务的配置结构。

cy_stc_ble_cgms_config_t cy_ble_cgmsConfig
连续血糖监测服务的配置结构。

cy_stc_ble_cps_config_t cy_ble_cpsConfig
循环供电服务的配置结构。

cy_stc_ble_cscs_config_t cy_ble_cscsConfig
循环速度和节奏服务的配置结构。

cy_stc_ble_cts_config_t cy_ble_ctsConfig
当前时间服务的配置结构。

cy_stc_ble_custom_config_t cy_ble_customConfig
自定义服务的配置结构。

cy_stc_ble_dis_config_t cy_ble_disConfig
器件信息服务的配置结构。

cy_stc_ble_ess_config_t cy_ble_essConfig
环境感应服务的配置结构。

cy_stc_ble_gls_config_t cy_ble_glsConfig
血糖监测服务的配置结构。

cy_stc_ble_hids_config_t cy_ble_hidsConfig
HID服务的配置结构。

cy_stc_ble_hps_config_t cy_ble_hpsConfig
HTTP代理服务的配置结构。

cy_stc_ble_hrs_config_t cy_ble_hrsConfig
心率服务的配置结构。

cy_stc_ble_hts_config_t cy_ble_htsConfig
健康温度计服务的配置结构。

cy_stc_ble_ias_config_t cy_ble_iasConfig
即时报警服务的配置结构。

cy_stc_ble_ips_config_t cy_ble_ipsConfig
室内定位服务的配置结构。

cy_stc_ble_lls_config_t cy_ble_llsConfig
链路损耗服务的配置结构。

cy_stc_ble_lns_config_t cy_ble_lnsConfig
定位和导航服务的配置结构。

cy_stc_ble_ndcs_config_t cy_ble_ndcsConfig
下个DST更改的服务的配置结构。

cy_stc_ble_pass_config_t cy_ble_passConfig
手机警报状态服务的配置结构。

cy_stc_ble_plxs_config_t cy_ble_plxsConfig
脉搏血氧计服务的配置结构。

cy_stc_ble_rscs_config_t cy_ble_rscsConfig
运行速度和节奏服务的配置结构。

cy_stc_ble_rtus_config_t cy_ble_rtusConfig
参考时间更新服务的配置结构。

cy_stc_ble_scps_config_t cy_ble_scpsConfig
扫描参数服务的配置结构。

cy_stc_ble_tps_config_t cy_ble_tpsConfig
Tx功耗服务的配置结构。

cy_stc_ble_uds_config_t cy_ble_udsConfig
用户数据服务的配置结构。

cy_stc_ble_wpts_config_t cy_ble_wptsConfig
无线电源传输服务的配置结构。

cy_stc_ble_wss_config_t cy_ble_wssConfig
计量秤服务的配置结构。

预处理器宏

BLE_PDL 组件生成以下预处理器宏。请注意，实际宏值取决于 Configure 对话框中“组件”的设置情况。

#define CY_BLE_CONFIG_MODE
BLE工作模式。

#define CY_BLE_CONFIG_HOST_CORE
配置文件模式下的主机内核。

#define CY_BLE_CONFIG_HCI_CONTR_CORE
HCI模式下的控制器内核。

#define CY_BLE_CONFIG_STACK_MODE
BLE协议栈内核模式：
- CY_BLE_STACK_MODE_SINGLE_SOC — 主机和控制器使用软件接口。
- CY_BLE_STACK_MODE_DUAL_IPC — 主机和控制器使用 IPC 接口。

#define CY_BLE_CONFIG_CONN_COUNT
最多BLE连接数。取值范围为1到4。

#define CY_BLE_CONFIG_GATTC_COUNT
BLE连接数（客户端）。

#define CY_BLE_CONFIG_GAP_PERIPHERAL_COUNT
GAP外设配置结构数量。

#define CY_BLE_CONFIG_GAP_BROADCASTER_COUNT
GAP广播器配置结构数。

#define CY_BLE_CONFIG_GAP_CENTRAL_COUNT
GAP中心器件配置结构数。

#define CY_BLE_CONFIG_GAP_OBSERVER_COUNT
GAP观察器配置结构数。

#define CY_BLE_CONFIG_AUTH_INFO_COUNT
安全配置结构数。

#define CY_BLE_CONFIG_AUTO_POPULATE_WHITELIST
提供一个将白名单链接到绑定器件列表的选项。

#define CY_BLE_CONFIG_MAX_RESOLVABLE_DEVICES
该器件应解析地址的最大对端器件数。

#define CY_BLE_CONFIG_ENABLE_LL_PRIVACY
LL保密1.2特性。

#define CY_BLE_CONFIG_ENABLE_PHY_UPDATE
LE 2 Mbps特性。

#define CY_BLE_CONFIG_GAP_SECURITY_LEVEL
GAP安全级别。

#define CY_BLE_CONFIG_GAP_ROLE
GAP模式。

#define CY_BLE_CONFIG_BONDING_REQUIREMENT
绑定要求。

#define CY_BLE_CONFIG_GATT_MTU
GATT MTU大小。

#define CY_BLE_CONFIG_GATT_DB_MAX_VALUE_LEN
最大GATT属性长度。

#define CY_BLE_CONFIG_GATT_RELIABLE_CHAR_COUNT
支持可靠写入属性的特性数量。

#define CY_BLE_CONFIG_GATT_RELIABLE_CHAR_LENGTH
具有可靠写入属性的特性总长度。

#define CY_BLE_CONFIG_GATT_ENABLE_EXTERNAL_PREP_WRITE_BUFF
该参数允许应用为一个写入请求准备一个动态分配缓冲区。

#define CY_BLE_CONFIG_L2CAP_ENABLE
该参数用于启用L2CAP逻辑通道的配置。

#define CY_BLE_CONFIG_L2CAP_MTU
L2CAP MTU大小。

#define CY_BLE_CONFIG_L2CAP_MPS
L2CAP MPS大小。

#define CY_BLE_CONFIG_L2CAP_LOGICAL_CHANNEL_COUNT
L2CAP逻辑通道数。

#define CY_BLE_CONFIG_L2CAP_PSM_COUNT
L2CAP PSMs数量。

#define CY_BLE_CONFIG_LL_MAX_TX_PAYLOAD_SIZE
最大Tx有效载荷大小。

#define CY_BLE_CONFIG_LL_MAX_RX_PAYLOAD_SIZE
最大Rx有效载荷大小。

#define CY_BLE_CONFIG_GATT_ROLE
GATT模式。

#define CY_BLE_CONFIG_HIDSS_SERVICE_COUNT
GATT服务器模式的最大支持HID服务数量。

#define CY_BLE_CONFIG_HIDSS_REPORT_COUNT
GATT服务器模式的最大支持HID报告数。

#define CY_BLE_CONFIG_HIDSC_SERVICE_COUNT
GATT客户端模式的最大支持HID服务数。

#define CY_BLE_CONFIG_HIDSC_REPORT_COUNT
GATT客户端模式的最大支持HID报告数。

#define CY_BLE_CONFIG_BASS_SERVICE_COUNT
GATT服务器模式的最大支持BAS服务数。

#define CY_BLE_CONFIG_BASC_SERVICE_COUNT
GATT客户端模式的最大支持BAS报告数。

#define CY_BLE_CONFIG_ES_TOTAL_CHAR_COUNT
GATT客户端模式的最大支持ESS特性数。

#define CY_BLE_CONFIG_AIO_TOTAL_CHAR_COUNT
GATT客户端模式的最大支持AIOS特性数。

#define CY_BLE_CONFIG_CUSTOMS_SERVICE_COUNT
GATT服务器模式的最大支持自定义服务数。

#define CY_BLE_CONFIG_CUSTOMC_SERVICE_COUNT
GATT客户端模式的最大支持自定义服务数。

#define CY_BLE_CONFIG_CUSTOM_SERVICE_CHAR_COUNT

最大支持的自定义服务特性数量。

#define CY_BLE_CONFIG_CUSTOM_SERVICE_CHAR_DESCRIPTORS_COUNT

一个特性中最大支持的自定义服务描述符数。

组件函数

该组件还包括一组组件特定的函数，从而简化了对基本 BLE 中间件操作的访问。

cy_en_ble_api_result_t Cy_BLE_Start (cy_ble_callback_t *callbackFunc*)

该函数初始化BLE协议栈，该协议栈包含BLE协议栈管理器、BLE控制器以及BLE主机模块。它还初始化BLE_PDL组件所要求的配置文件层、调度器、定时器以及其它平台相关的资源。此外，该参数还注册BLE协议栈中BLE事件的回调函数。

请注意，该函数并不复位BLE协议栈。

在HCI操作模式下，该函数将不初始化BLE主机模式。

当成功初始化BLE协议栈时，通过调用该函数，可以生成CY_BLE_EVT_STACK_ON事件。

在双核模式下，在两个内核上应按以下顺序调用该函数：

- 在CM0+上调用该函数，以便初始化控制器。
- 调用Cy_SysEnableCM4()函数，启动CM4内核。
- 在CM4上调用该函数以便初始化主机和配置文件。

参数：

callbackFunc	用于接收来自BLE协议栈事件的回调函数。 cy_ble_callback_t是函数指针类型。

返回值：

cy_en_ble_api_result_t：返回值表示函数执行的成败。下面是可能发生的错误代码。

错误代码	描述
CY_BLE_SUCCESS	成功操作。
CY_BLE_ERROR_INVALID_PARAMETER	未在HCI模式下编译BLE协议栈时，会将NULL指针传递给函数。在HCI模式下，永远不会返回CY_BLE_ERROR_INVALID_PARAMETER。
CY_BLE_ERROR_REPEATED_ATTEMPTS	多次调用该函数，但各次调用间隔内不调用Cy_BLE_StackShutdown()函数。
CY_BLE_ERROR_MEMORY_ALLOCATION_FAILED	内存不足。

全局变量

cy_ble_initVar变量用于说明该组件的初始配置。该变量被初始化为0（0u）并在第一次调用Cy_BLE_Start()时设置为1（1u）。这样允许进行组件初始化，而无需在所有对Cy_BLE_Start()子程序的后续调用中再次初始化。

cy_en_ble_api_result_t Cy_BLE_Stop (void)

该函数停止BLE协议栈中正在进行的任意操作，并强制BLE协议栈关闭。在Cy_BLE_GetState() API返回CY_BLE_STATE_STOPPED state的场合，调用该函数后只能调用Cy_BLE_Start()函数。

成功关闭协议栈时，调用该函数会引起发生CY_BLE_EVT_STACK_SHUTDOWN_COMPLETE事件。

返回值：

cy_en_ble_api_result_t：返回值表示函数执行成功还是失败。下面是可能发生的错误代码。

错误代码	描述
CY_BLE_SUCCESS	成功操作。
CY_BLE_ERROR_INVALID_OPERATION	调用Cy_BLE_Start前便调用了Cy_BLE_Stop。

回调函数

BLE_PDL 组件要求您定义一个用于处理 BLE 协议栈事件的回调函数。该函数将作为一个参数传送给 Cy_BLE_Start() API。回调函数的类型为 cy_ble_callback_t，如下定义：

```
void (* cy_ble_callback_t)(uint32_t eventCode, void *eventParam);
```

- eventCode：协议栈事件代码
- eventParam：协议栈事件参数

回调函数应该评估 eventCode（和用于某些事件的 eventParam）并提供特定于协议栈事件的操作。因此，在一般事件中（如广播、扫描、连接和超时），使用这些事件来编译您的应用特定状态机。

在调用 Cy_BLE_ProcessEvents() API 方法之后，BLE 协议栈会触发应用事件处理程序回调，用于处理链接层生成的所有挂起事件。但用于请求主机生成操作的其它 BLE_PDL 组件 API 也会触发应用事件处理程序回调函数，从而可以在返回这些 API 方法前处理事件。

同样，您应为所需要的每一个服务提供一个回调函数。该函数的类型为 cy_ble_callback_t 并作为参数传送到服务特定的回调注册函数。该回调函数用于评估服务特定事件，从而执行您的应用程序所定义的相应操作。此时，可以使用这些事件构建服务特定的状态机。

代码段

- 对于应用程序回调：void Cy_BLE_AppCallback(uint32_t eventCode, void *eventParam){<所有通用事件>}
- 对于每个 Cy_BLE_<service>RegisterAttrCallback API 函数：Cy_BLE_<service>RegisterAttrCallback(Cy_BLE_<service>CallBack);
- 对于每个服务回调：void Cy_BLE_<service>CallBack(uint32_t eventCode, void *eventParam) {<所有服务特定的事件>}

代码示例和应用笔记

代码示例

在 Find Example Project（查找示例项目）对话框中，PSoC Creator 提供了大量的代码示例，包括原理图和示例代码。要查看组件特定示例，请打开“Component Catalog”（组件目录）中的对话框或原理图中的组件实例。要查看通用示例，请打开“Start Page”（起始页）或 **File**（文件）菜单中的对话框。根据要求，可以通过使用对话框中的 **Filter Options**（过滤选项）项来限定可选的项目列表。

更多有关信息，请参考 PSoC Creator 帮助中主题为“Find Code Example”（查找代码示例）的部分内容。

应用笔记

赛普拉斯提供了许多应用笔记，介绍了如何将 PSoC 集成到您的设计中。您可以在赛普拉斯应用笔记搜索网页 www.cypress.com/appnotes 上访问这些文档。使用该组件的应用笔记包括：

- AN210781 — PSoC 6 BLE 入门手册
- AN95089 — PSoC/PRoC BLE 晶体振荡器选择和调校技术
- AN91445 — 天线设计和射频布局指南

工业标准

MISRA 合规性

本节介绍了 MISRA-C:2004 合规性和本组件的偏差情况。有三种差异类型，如下定义：

- 项目偏差 — 适用于所有 PSoC Creator 组件的偏差
- 组件特定偏差 — 仅适用于该组件的通用部分的偏差
- 配置文件特定偏差 — 仅适用于组件的特定配置文件的偏差有关配置文件特定偏差的详细说明，请参阅 BLE 中间件库文档。

本节提供了有关组件特定偏差的信息。PSoC Creator Help（PSoC Creator 帮助）> Building a PSoC Creator Project （构建一个 PSoC Creator 项目）> Generated Files（生成文件）（PSoC 6）中的 MISRA 合规性部分介绍了项目偏差以及有关 MISRA 合规性验证环境的信息。

BLE_PDL 组件具有以下各项特定偏差。

MISRA-C:2004 规则	规则类别（必须（R）/ 建议（A））	规则说明	偏差说明
1.1	R	该规则规定，代码必须符合 C ISO/IEC 9899:1990标准。	该组件支持ISO:C99标准。
1.2	R	不应依赖于未定义或未指定的行为。	规则5.1中明确介绍了该特定行为。
3.1	A	不应对指向易失性对象的指针和一种整数类型进行转换。	从无符号整数到指针的转换不会造成任何意外影响，因为它是基于对硬件寄存器结构定义的结果。
5.1	R	内部和外部标示符都不应依赖多于31个字符的有效性。	该规则适用于ISO:C90标准。符合ISO:C99标准的组件特定代码不使用该限制。
10.1	R	在某些情况下，整数类型表达式的数值不应被完全转换为不同的底层类型。	进行算术或条件操作后，枚举类型的操作数将被转换到无符号类型。该转换过程不会引起任何意外效果。
11.4	A	指向对象类型的不同指针之间不应进行转换。	在指针间进行转换时，必须保证这些指针正确对准了所指向的对象类型。

该器件配有以下嵌入式构件：cy_isr、SCB。有关 MISRA 合规性与特定偏差的相关信息，请参见相应组件数据手册。

蓝牙认证

赛普拉斯提供的 BLE 解决方案在蓝牙 SIG 网站上被列为经过认证的解决方案。该认证是模块式的，所以用户可以更加灵活地使用。以下是合格的设计 ID（QD ID）和声明 ID 的列表。

QD ID	声明ID	说明
99158	D037716	具有蓝牙低功耗（BLE）连接功能的PSoC 6 MCU可为物联网（IoT）提供超低功耗及同类中最佳的灵活内置安全性。它基于超低功耗40 nm工艺技术，并采用了双核ARM® Cortex®-M架构，包括Cortex-M4F和Cortex-M0 +。BLE无线电采用了超低功耗55 nm的工艺制造。具有BLE连接功能的PSoC 6 MCU将BLE子系统以及软件定义的模拟和数字外设、CapSense®、可编程互连、高性能双核架构和关键的安全功能集成在一个单芯片中。

使用资源

BLE_PDL 组件使用了一个 BLESS 模块、两个外部晶振、（多个）中断以及一个可选 SCB 模块：

配置	资源类型				
	BLESS[1]	SCB [2]	中断	ECO	WCO [3]
配置文件模式	1	-	1	1	1
HCI模式	1	1	2	1	1

直流和交流的电气特性

除非另有说明，否则这些规范的适用条件是：–40°C ≤ T_A ≤ 85 °C 且 T_J ≤ 100 °C，电压范围为 1.71 V ~ 5.5 V。

注意：目前，PSoC 6 器件的最终版特征数据尚未得出。数据可用时，将在赛普拉斯网站上更新组件数据手册。

参数	说明	最小值	典型值	最大值	单位	详情/条件
BLE 子系统规范						
射频接收器规范（1 Mbps）						
RXS、IDLE	发送器空闲时的 RX 灵敏度	–	–95	–	dBm	全射频工作频率范围
RXS、IDLE	发送器空闲时的 RX 灵敏度	–	–93	–	dBm	255 字节的数据包长度、全频率范围
RXS、DIRTY	发送器有扰时的 RX 灵敏度	–	–92	_	dBm	RF-PHY 规范（RCV-LE/CA/01/C）
PRXMAX	最大接收信号强度（< 0.1% PER 时）	–	0	–	dBm	RF-PHY 规范（RCV-LE/CA/06/C）
CI1	同通道干扰，所需信号强度为 –67 dBm，干扰器频率为 FRX	–	9	21	dB	RF-PHY 规范（RCV-LE/CA/03/C）
CI2	相邻通道干扰。所需信号强度为 –67 dBm，干扰器频率为 FRX ± 1 MHz	–	3	15	dB	RF-PHY 规范（RCV-LE/CA/03/C）
CI3	相邻通道干扰。所需信号强度为 –67 dBm，干扰器频率为 FRX ± 2 MHz	–	–29	–17	dB	RF-PHY 规范（RCV-LE/CA/03/C）

[1] 在 HCI 模式下进行配置时，BLESS 组件会实例化一个 SCB 组件。有关其资源使用情况，请参见 SCB 组件数据手册。

[2] 在 HCI 模式下进行配置时，BLE 组件会实例化 SCB 组件。有关其资源使用情况，请参见 SCB 组件数据手册。

[3] WCO 是可选的。需要组件深度睡眠模式时，会使用它。如果不使用 WCO，则 ILO 作为 LFCLK 时钟原使用。

参数	说明	最小值	典型值	最大值	单位	详情/条件
CI4	相邻通道干扰。所需信号强度为–67 dBm，干扰器频率为 FRX ± 3 MHz	–	–39	–27	dB	RF-PHY 规范（RCV-LE/CA/03/C）
CI5	相邻通道干扰。所需信号强度为–67 dBm，干扰器频率为镜像频率（FIMAGE）	–	–20	–9	dB	RF-PHY 规范（RCV-LE/CA/03/C）
CI6	相邻通道干扰。所需信号强度为–67 dBm，干扰器频率为镜像频率（FIMAGE ± 1 MHz）	–	–30	–15	dB	RF-PHY 规范（RCV-LE/CA/03/C）
射频接收器规范（2 Mbps）						
RXS、IDLE	发送器空闲时的 RX 灵敏度		–92		dBm	全射频工作频率范围
RXS、IDLE	发送器空闲时的 RX 灵敏度		–90		dBm	255 字节的数据包长度、全频率范围
RXS、DIRTY	发送器有扰时的 RX 灵敏度		–89		dBm	RF-PHY 规范（RCV-LE/CA/01/C）
PRXMAX	最大接收信号强度（< 0.1% PER 时）		0		dBm	RF-PHY 规范（RCV-LE/CA/06/C）
CI1	同通道干扰，所需信号强度为–67 dBm，干扰器频率为 FRX		9	21	dB	RF-PHY 规范（RCV-LE/CA/03/C）
CI2	相邻通道干扰，所需信号强度为–67 dBm，干扰器频率为 FRX ± 2 MHz		3	15	dB	RF-PHY 规范（RCV-LE/CA/03/C）
CI3	相邻通道干扰，所需信号强度为–67 dBm，干扰器频率为 FRX ± 4 MHz		–29	–17	dB	RF-PHY 规范（RCV-LE/CA/03/C）
CI4	相邻通道干扰，所需信号强度为–67 dBm，干扰器频率为 FRX ± 6 MHz		–39	–27	dB	RF-PHY 规范（RCV-LE/CA/03/C）
CI5	相邻通道干扰，所需信号强度为–67 dBm，以及干扰器频率为镜像频率（FIMAGE）		–20	–9	dB	RF-PHY 规范（RCV-LE/CA/03/C）
CI6	相邻通道干扰，所需信号强度为–67 dBm，干扰器频率为（FIMAGE ± 2 MHz）		–30	–15	dB	RF-PHY 规范（RCV-LE/CA/03/C）
射频接收器规范（1 和 2 Mbps）						
OBB1	频带外封锁，所需信号强度为–67 dBm，干扰器的频率范围为 30 ~ 2000 MHz	–30	–27	–	dBm	RF-PHY 规范（RCV-LE/CA/04/C）
OBB2	频带外封锁，所需信号强度为–67 dBm，干扰器的频率范围为 2003 ~ 2399 MHz	–35	–27		dBm	RF-PHY 规范（RCV-LE/CA/04/C）

参数	说明	最小值	典型值	最大值	单位	详情/条件
OBB3	频带外封锁，所需信号强度为 –67 dBm，干扰器的频率范围为 2484 ~ 2997 MHz	–35	–27		dBm	RF-PHY 规范（RCV-LE/CA/04/C）
OBB4	频带外封锁，所需信号强度为 –67 dBm，干扰器的频率范围为 3000 ~ 12750 MHz	–30	–27		dBm	RF-PHY 规范（RCV-LE/CA/04/C）
IMD	互调性能所需的信号强度为 –64 dBm，1 Mbps BLE，第三、第四和第五个偏移通道	–50			dBm	RF-PHY 规范（RCV-LE/CA/05/C）
RXSE1	接收器发散发射频率范围为 30 MHz ~ 1.0 GHz			–57	dBm	100 kHz 测量带宽，符合 ETSI EN300 328 版本 1.8.1
RXSE2	接收器发散发射频率范围为 1.0 GHz ~ 12.75 GHz			–54	dBm	1 MHz 测量带宽，符合 ETSI EN300 328 版本 1.8.1
RF 发送器规范						
TXP、ACC	射频功率精度	–1	–	1	dB	–
TXP、RANGE	频率精度		24		dB	–20 dBm ~ +4 dBm
TXP、0dBm	输出功率，0 dB 增益设置		0		dBm	–
TXP、MAX	输出功率，最大功率设置		4		dBm	–
TXP、MIN	输出功率，最小功率设置		–20		dBm	–
F2AVG	10101010 格式的平均频率偏差	185			kHz	RF-PHY 规范（TRM-LE/CA/05/C）
F2AVG_2M	数据速率为 2 Mpbs 时，10101010 格式的平均频率偏差	370			kHz	RF-PHY 规范（TRM-LE/CA/05/C）
F1AVG	11110000 格式的平均频率偏差	225	250	275	kHz	RF-PHY 规范（TRM-LE/CA/05/C）
F1AVG_2M	数据速率为 2 Mpbs 时，11110000 格式的平均频率偏差	450	500	550	kHz	RF-PHY 规范（TRM-LE/CA/05/C）
EO	眼开程度 = ΔF2AVG/ΔF1AVG	0.8				RF-PHY 规范（TRM-LE/CA/05/C）
FTX、ACC	频率精度	-150		150	kHz	RF-PHY 规范（TRM-LE/CA/06/C）
FTX、MAXDR	最大频率漂移	–50		50	kHz	RF-PHY 规范（TRM-LE/CA/06/C）
FTX、INITDR	初始频率漂移	–20		20	kHz	RF-PHY 规范（TRM-LE/CA/06/C）
FTX、DR	最大漂移率	–20		20	kHz/ 50 µs	RF-PHY 规范（TRM-LE/CA/06/C）
IBSE1	带内发散发射（偏移为 2 MHz，数据速率为 1 Mbps） 带内发散发射（偏移为 4 MHz，数据速率为 2 Mbps）			–20	dBm	RF-PHY 规范（TRM-LE/CA/03/C）

参数	说明	最小值	典型值	最大值	单位	详情/条件
IBSE2	带内发散发射（偏移 ≥ 3 MHz，数据速率为 1 Mbps） 带内发散发射（偏移 ≥ 6 MHz，数据速率为 2 Mbps）			–30	dBm	RF-PHY 规范（TRM-LE/CA/03/C）
TXSE1	发送器发散发射（平均），< 1.0 GHz			–55.5	dBm	FCC-15.247
TXSE2	发送器发散发射（平均），> 1.0 GHz			–41.5	dBm	FCC-15.247
射频电流规范						
IRX1_wb	接收电流（1 Mbps）		4.4		mA	使用降压转换器时的 AVIN/PVIN 和 VIO 电流
ITX1_wb_0dBm	功率设置为 0 dBm 时的 TX 电流（1 Mbps）		4.2		mA	使用降压转换器时的 AVIN/PVIN 和 VIO 电流（AVIN/PVIN = 1.8 V、VIO = 1.8 V）
IRX1_nb	接收电流（1 Mbps）		9.5		mA	不使用降压转换器时的 AVIN/PVIN 和 VIO 电流（AVIN/PVIN = 1.8 V、VIO = 1.8 V）
ITX1_nb_0dBm	功率设置为 0 dBm 时的 TX 电流（1 Mbps）		9		mA	不使用降压转换器时的 AVIN/PVIN 和 VIO 电流（AVIN/PVIN = 1.8 V、VIO = 1.8 V）
ITX1_nb_4dBm	功率设置为 4 dBm 时的 TX 电流（1 Mbps）		13		mA	不使用降压转换器时的 AVIN/PVIN 和 VIO 电流（AVIN/PVIN = 1.8 V、VIO = 1.8 V）
ITX1_wb_4dBm	功率设置为 4 dBm 时的 TX 电流（1 Mbps）		6.5		mA	使用降压转换器时的 AVIN/PVIN 和 VIO 电流（AVIN/PVIN = 1.8 V、VIO = 1.8 V）
ITX1_nb_20dBm	功率设置为-20 dBm 时的 TX 电流（1 Mbps）		7		mA	不使用降压转换器时的 AVIN/PVIN 和 VIO 电流（AVIN/PVIN = 1.8 V、VIO = 1.8 V）
IRX2_wb	接收电流（2 Mbps）		4.4		mA	使用降压转换器时的 AVIN/PVIN 和 VIO 电流（AVIN/PVIN = 1.8 V、VIO = 1.8 V）
ITX2_wb_0dBm	功率设置为 0 dBm 时的 TX 电流（2 Mbps）		4.2		mA	使用降压转换器时的 AVIN/PVIN 和 VIO 电流（AVIN/PVIN = 1.8 V、VIO = 1.8 V）
IRX2_nb	接收电流（2 Mbps）		9.5		mA	不使用降压转换器时的 AVIN/PVIN 和 VIO 电流（AVIN/PVIN = 1.8 V、VIO = 1.8 V）
ITX2_nb_0dBm	功率设置为 0 dBm 时的 TX 电流（2 Mbps）		9		mA	不使用降压转换器时的 AVIN/PVIN 和 VIO 电流（AVIN/PVIN = 1.8 V、VIO = 1.8 V）

参数	说明	最小值	典型值	最大值	单位	详情/条件
ITX2_nb_4dBm	功率设置为 4 dBm 时的 TX 电流（2 Mbps）		13		mA	不使用降压转换器时的 AVIN/PVIN 和 VIO 电流（AVIN/PVIN = 1.8 V、VIO = 1.8 V）
ITX2_wb_4dBm	功率设置为 4 dBm 时的 TX 电流（2 Mbps）		6.5		mA	使用降压转换器时的 AVIN/PVIN 和 VIO 电流（AVIN/PVIN = 1.8 V、VIO = 1.8 V）
ITX2_nb_20dBm	功率设置为–20 dBm 时的 TX 电流（2 Mbps）		7		mA	不使用降压转换器时的 AVIN/PVIN 和 VIO 电流（AVIN/PVIN = 1.8 V、VIO = 1.8 V）
通用射频规范						
FREQ	射频工作频率	2400		2482	MHz	–
CHBW	通道间距		2		MHz	–
DR1	无线传输数据速率（1 Mbps）		1000		Kbps	–
DR2	无线传输数据速率（2 Mbps）		2000		Kbps	–
TXSUP	发送器启动时间		80	82	s	–
RXSUP	接收器启动时间		80	82	s	–
RSSI 规范						
RSSI、ACC	RSSI 精度	–4	–	4	dB	–
RSSI、RES	RSSI 分辨率		1		dB	–
RSSI、PER	RSSI 采样周期		6		s	–
系统级 BLE 规范						
Adv_Pwr	1.28 s、32 字节、0 dBm		21		µW	3.3 V 电压、降压、无深度睡眠电流
Conn_Pwr_300	300 ms、0 字节、0 dBm		33		µW	3.3 V 电压、降压、无深度睡眠电流
Conn_Pwr_1S	1000 ms、0 字节、0 dBm		10		µW	3.3 V 电压、降压、无深度睡眠电流
Conn_Pwr_4S	4000 ms、0 字节、0 dBm		3		µW	3.3 V 电压、降压、无深度睡眠电流

下列表格汇总了 BLE 固件协议栈执行/启动不同 BLE 操作所使用时间的测量情况。这里使用了 IMO 执行测量操作，其中测量条件如下：FLL 被设置为 24 MHz，VDD 被设置为 1.1 V，连接间隔被设置为 7.5 ms，并且加密功能被启用。

操作	时长（µs）
BLE协议栈启动时间（无DLE、链路层保密、2 Mbps以及安全连接）	21032
BLE协议栈启动时间（有DLE、链路层保密、2 Mbps以及安全连接）	21137

操作	时长（µs）
BLE协议栈关闭时间	1170
CLOSE_CE ISR执行时间 — SSSS配置	310
CLOSE_CE ISR执行时间 — MMMM配置	254
EARLY_INTR ISR执行时间 — SSSS配置	138
EARLY_INTR ISR执行时间 — MMMM配置	84
启动扫描执行时间	680
ADV数据包处理时间（ADV数据包的长度 — 31字节）	212
SCAN_RESPONSE数据包处理时间（扫描响应数据长度 — 31字节）	218
GAP中心器件上持续的时间	355824
GAP外设上持续的时间	780
启动广播执行时间（最坏情况）	1568
广播数据更新时间	368
没有DLE（1片段）的20字节GATT服务器上的通知处理时间	635
没有DLE（3片段）的74字节GATT服务器上的通知处理时间	1683
有DLE（1片段）的244字节GATT服务器上的通知处理时间	885
有DLE（2片段）的495字节GATT服务器上的通知处理时间	2340
没有DLE（1片段）的20字节GATT客户端上的通知处理时间	480
有DLE（1片段）的244字节GATT客户端上的通知处理时间	806

BLE 协议栈更改

本节列出了对 BLE 协议栈进行的更改。

版本	更改说明	更改原因/影响
5.0.0.892	如果要安排的事件已经成为过去式，则更新BLE协议栈以便在关闭CE时重新运行调度程序。	阻挡的闪存写操作会阻止CPU执行任何中断（包括BLESS isr）超过17 ms时长。因此，如果连接的CI小于17 ms，则BLE链路将被断开连接。
	在用户定义的条件下修改最大输出功率级别的校准目标值。	增加这项功能是为了支持 BLE 协议栈中的 5 dBm Tx 功率。
5.0.0.855	将Cy_BLE_SetCustomEventMask() API的输入参数从(uint8* mask)更改为(uint32 mask)。	与位掩码机制达成一致。

版本	更改说明	更改原因/影响
	1. 定时器上下文不会被复位，它仅在协议栈关闭期间被重置。 2. 即使计时器启动失败，也不会释放计时器上下文。由于其他原因，当计时器启动失败时，它被释放。	通过调用Cy_BLE_GAPC_StopScan()函数来停止先前启动的扫描过程时，Cy_BLE_GAPC_StopScan()函数会返回CY_BLE_ERROR_MAX。 只有在身份验证过程发生并且应用程序尝试停止扫描时，才会出现该问题。
	1. 只有硬件FIFO中的排队数据长度请求成功时，才会启动DLE控制过程计时器。 2. 增加检查操作，以处理来自主机的ACK处理和处理数据PDU之间的优先条件。	由于嵌入式载荷应用程序中的LMP响应超时而导致意外断开连接。原因在于： 1. 即使硬件FIFO中的排队数据长度请求发送失败，仍然启动DLE控制过程计时器。 2. 来自主机的ACK处理和处理数据PDU之间的优先条件。
5.0.0.785	初始BLE协议栈版本。	

组件更改

本节列出了该组件在各版本中主要的更改内容。

版本	更改说明	更改原因/影响
2.0.c	添加了BLE用户配置文件*cy_ble_config.h*。	使BLE配置更加灵活。 允许重新定义由BLE定制器所生成的配置定义和默认的BLE时钟定义（来自*cy_ble_clk.h*）。
	更新了BLE中断通知功能。	增加了对BLE双线模式的支持。
	更新了以下函数： Cy_BLE_GATTS_ReadAttributeValueLocal() Cy_BLE_GATTS_ReadAttributeValuePeer()	这些函数具有错误的本地/对端启动操作标志。
	不推荐使用CY_BLE_CONFIG_STACK_MODE配置参数的以下定义值： CY_BLE_CONFIG_STACK_HOST_ONLY CY_BLE_CONFIG_STACK_DEBUG_UART CY_BLE_CONFIG_STACK_HOST_IPC.	不推荐使用不受支持（调试）的模式。
	重新命名以下定义值： 将CY_BLE_CONFIG_STACK_DEBUG重命名为CY_BLE_STACK_MODE_SINGLE_SOC 将CY_BLE_CONFIG_STACK_RELEASE重命名为CY_BLE_STACK_MODE_DUAL_IPC。	重新命名CY_BLE_CONFIG_STACK_DEBUG和CY_BLE_CONFIG_STACK_RELEASE的定义，使其名称更有意义。

版本	更改说明	更改原因/影响
	更新了数据手册。	更新了“BLE中断通知回调”章节。 更新了“快速入门”章节。添加了关于同Em_EEPROM配对使用BLE的说明。 删除了“组件勘误表”章节。
	将BLE协议栈更新为5.0.0.892版本。	请参见BLE协议栈更改部分。
2.0.b	更新了数据手册。	添加了“多连接支持”章节。 添加了“BLE中断通知回调”章节。 更新了直流和交流电气特性章节。 为通过代码共享特性进行无线引导加载添加了说明。
	添加了新API： • Cy_BLE_IsDevicePaired()、 • Cy_BLE_GetDeviceRole()、 • Cy_BLE_IsPeerConnected()、 • Cy_BLE_GATTS_WriteAttributeValuePeer()、 • Cy_BLE_GATTS_WriteAttributeValueLocal()、 • Cy_BLE_GATTS_ReadAttributeValuePeer()、 • Cy_BLE_GATTS_ReadAttributeValueLocal()、 • Cy_BLE_GATTS_IsIndicationEnabled()、 • Cy_BLE_GATTS_SendIndication()、 • Cy_BLE_GATTS_IsNotificationEnabled()。 • Cy_BLE_GATTS_SendNotification()、 • Cy_BLE_GATTS_SendErrorRsp()、 • Cy_BLE_GATTC_SendConfirmation()	增加可用性。
	将BLE协议栈更新为5.0.0.855版本。	请参见BLE协议栈更改部分。
2.0.a	更新了数据手册。	添加了新配置文件的说明：自动化IO和脉搏血氧计。 更新了“快速入门”章节。 更新了“MISRA合规性”章节。
	为CPU内核参数添加新选项。	支持单核器件。
	链路层参数已被移至定制器中的单独选项卡内。	更新屏幕；添加了“链路层设置”的说明
	使最大连接数量具有可配置性。	不使用额外连接时，SRAM的占用空间被减少。

版本	更改说明	更改原因/影响
	重新命名以下API： • 将Cy_BLE_GAP_UpdateAdvScanData()重命名为Cy_BLE_GAPP_UpdateAdvScanData()； • 将Cy_BLE_SetConnectionPriority()重命名为Cy_BLE_GAP_SetConnectionPriority().	使API命名机制统一。
	将BLE协议栈更新为5.0.0.785版本。	请参见BLE协议栈更改部分。
2.0	更新了底层PDL驱动程序版本。 更新了数据手册。	更新屏幕；添加了“链路层设置”的说明
1.0.a	更新了数据手册。	使有关IAR编译器的信息更加清楚。 为通用检测模式添加注释。 添加了“快速入门”章节。
1.0	BLE组件PDL版本的最初文档。	
	初始BLE协议栈版本5.0.0。	

© 赛普拉斯半导体公司，2017-2019 年。本文件是赛普拉斯半导体公司及其子公司，包括 Spansion LLC（“**赛普拉斯**”）的财产。本文件，包括其包含或引用的任何软件或固件（“**软件**”），根据全球范围内的知识产权法律以及美国与其他国家签署条约由赛普拉斯所有。除非在本款中另有明确规定，赛普拉斯保留在该等法律和条约下的所有权利，且未就其专利、版权、商标或其他知识产权授予任何许可。如果软件并不附随有一份许可协议且贵方未以其他方式与赛普拉斯签署关于使用软件的书面协议，赛普拉斯特此授予贵方属人性质的、非独家且不可转让的如下许可（无再许可权）（1）在赛普拉斯特软件著作权项下的下列许可权（一）对以源代码形式提供的软件，仅出于在赛普拉斯硬件产品上使用之目的且仅在贵方集团内部修改和复制软件，和（二）仅限于在有关赛普拉斯硬件产品上使用之目的将软件以二进制代码形式的向外部最终用户提供（无论直接提供或通过经销商和分销商间接提供），和（2）在被软件（由赛普拉斯公司提供，且未经修改）侵犯的赛普拉斯专利的权利主张项下，仅出于在赛普拉斯硬件产品上使用之目的制造、使用、提供和进口软件的许可。禁止对软件的任何其他使用、复制、修改、翻译或汇编。

在适用法律允许的限度内，赛普拉斯未对本文件或任何软件作出任何明示或暗示的担保，包括但不限于关于适销性和特定用途的默示保证。没有任何电子设备是绝对安全的。因此，尽管赛普拉斯在其硬件和软件产品中采取了必要的安全措施，但是赛普拉斯并不承担任何由于使用赛普拉斯产品而引起的安全问题及安全漏洞的责任，例如未经授权的访问或使用赛普拉斯产品。此外，本材料中所介绍的赛普拉斯产品有可能存在设计缺陷或设计错误，从而导致产品的性能与公布的规 格不一致。（如果发现此类问题，赛普拉斯会提供勘误表）赛普拉斯保留更改本文件的权利，届时将不另行通知。在适用法律允许的限度内，赛普拉斯不对因应用或使用本文件所述任何产品或电路引起的任何后果负责。本文件，包括任何样本设计信息或程序代码信息，仅为供参考之目的提供。文件使用人应负责正确设计、计划和测试信息应用和由此生产的任何产品的功能和安全性。赛普拉斯产品不应被设计为、设定为或授权用作武器操作、武器系统、核设施、生命支持设备或系统、其他医疗设备或系统（包括急救设备和手术植入物）、污染控制或有害物质管理系统中的关键部件，或产品植入之设备或系统故障可能导致人身伤害、死亡或财产损失其他用途（“**非预期用途**”）。关键部件指，若该部件发生故障，经合理预期会导致设备或系统故障或会影响设备或系统安全性和有效性的部件。针对由赛普拉斯产品非预期用途产生或相关的任何主张、费用、损失和其他责任，赛普拉斯不承担全部或部分责任且贵方不应追究赛普拉斯之责任。贵方应赔偿赛普拉斯因赛普拉斯产品任何非预期用途产生或相关的所有索赔、费用、损失和其他责任，包括因人身伤害或死亡引起的主张，并使之免受损失。

赛普拉斯、赛普拉斯徽标、Spansion、Spansion 徽标，及上述项目的组合，WICED，及 PSoC、CapSense、EZ-USB、F-RAM 和 Traveo 应视为赛普拉斯在美国和其他国家的商标或注册商标。请访问 cypress.com 获取赛普拉斯商标的完整列表。其他名称和品牌可能由其各自所有者主张为该方财产。