PSoC 4 BLE Find Me

About this document

Scope and purpose

This code example demonstrates the implementation of a simple Bluetooth Low Energy (BLE) Immediate Alert Service (IAS)-based Find Me profile using PSoC® 4 MCU with BLE Connectivity.

Requirements

Tool: Tool: PSoC Creator 4.4, CySmart 1.3

Programming Language: C (Arm® GCC 5.4.1)

Associated Parts: PSoC 4 devices with BLE

Related Hardware: CY8CKIT-042-BLE-A Bluetooth Low Energy 4.2 Compliant Pioneer Kit

Table of contents

About this document **1**
Requirements **1**
Table of contents **1**
1 Objective **2**
2 Overview **3**
3 Hardware and Software Setup **4**
3.1 Hardware Setup 4
3.2 Software Setup 4
4 Operation **5**
4.1 Test Using the CySmart Mobile App 5
4.1.1 Test Using the 'Find Me' Profile 5
4.1.2 Test Using the 'GATT DB' Profile 6
4.2 Test Using the CySmart Host Emulation Tool 7
5 Design and Implementation **11**
5.1 BLE 11
6 Components and Settings **14**
7 Reusing This Example **18**
8 References **19**
9 Revision History **20**

1 Objective

This code example demonstrates the implementation of a simple Bluetooth Low Energy (BLE) Immediate Alert Service (IAS)-based Find Me profile using PSoC® 4 MCU with BLE Connectivity.

2 Overview

This design implements a Bluetooth Low Energy (BLE) Find Me Profile (FMP) that consists of an Immediate Alert Service (IAS). FMP and IAS are BLE standard Profile and Service respectively, as defined by the Bluetooth Special Interest Group (SIG). The design uses the RGB LED on the CY8CKIT-042-BLE-A Bluetooth Low Energy 4.2-Compliant Pioneer Kit. The blue LED displays the alert level (OFF, flashing, or ON for no alert, mild alert, or high alert respectively). Green and red LEDs indicate whether the BLE Peripheral (the Pioneer kit) is advertising or disconnected. The USB-BLE dongle provided with the CY8CKIT-042-BLE-A Pioneer kit or an iOS/Android mobile device can act as the BLE Central device, which locates the Peripheral device.

3 Hardware and Software Setup

3.1 Hardware Setup

This example uses the kit's default configuration. See the kit guide to ensure that the kit is configured correctly.

3.2 Software Setup

This code example consists of two parts: a locator and a target. For the locator, download and install either the **CySmart Host Emulation Tool** PC application or the CySmart app for **iOS** or **Android**. You can test the behavior with any of the two options, but the CySmart app is simpler.

Scan one of the following QR codes from your mobile phone to download the CySmart app.

iOS

Android

4 Operation

1. Plug the CY8CKIT-042-BLE-A kit board into your computer's USB port. The green LED turns ON to show that BLE is advertising. The BLE advertisement will time out after 30 seconds.
2. Build the project and program it into the PSoC 4 device. Choose **Debug** > **Program**. For more information on device programming, see PSoC Creator Help in the PSoC Creator toolbar.
3. Observe that the green LED turns ON to indicate device advertisement.

4.1 Test Using the CySmart Mobile App

4. Turn ON Bluetooth on your Android or iOS device.
5. Launch the CySmart app.
6. Press the "User" switch (SW2) on the Pioneer Kit to restart BLE advertisements. The advertisement LED (green LED) turns ON to indicate that BLE advertisement has started.
7. Pull down the CySmart app home screen to start scanning for BLE Peripherals; your device appears in the CySmart app home screen. Select your device "Find Me Target" to establish a BLE connection. Once the connection is established, the green LED turns OFF.

4.1.1 Test Using the 'Find Me' Profile

1. Select the 'Find Me' Profile from the carousel view.



Figure 1 Profile Selection

2. Select an Alert Level value on the Find Me **Profile** screen. Observe the state of the blue LED on the device change based on the alert level.

Alert Level	LED Status
No Alert	LED OFF
Mild Alert	Blinking
High Alert	LED ON

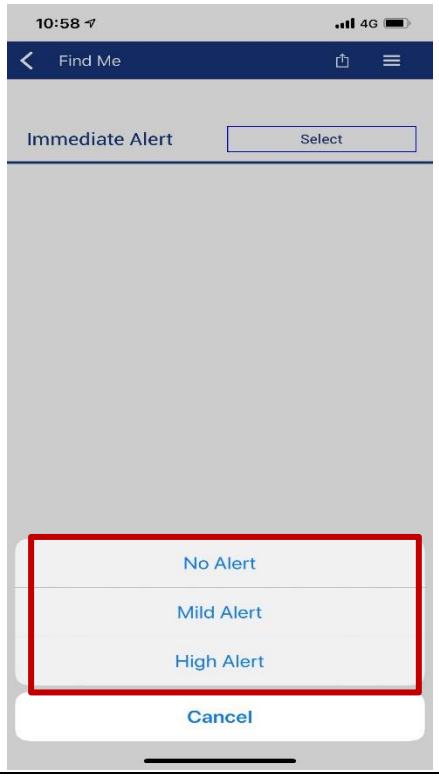


Figure 2 Alert Level Selection

4.1.2 Test Using the 'GATT DB' Profile

1. Select the 'GATT DB' profile from the carousel view.

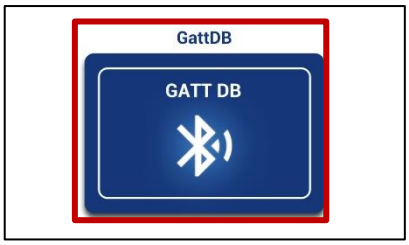


Figure 3 GattDB Profile Selection

2. Select the 'Immediate Alert' service.
3. Select the 'Alert Level' Characteristic.
4. Write a value of 0 (0x00) – no alert, 1 (0x01) – mid alert, or 2 (0x02) – high alert by entering the value in the 'HEX' input. Observe the LED for different alert levels.

Alert Level	HEX Value	LED Status
No Alert	0x00	OFF
Mild Alert	0x01	Blinking
High Alert	0x02	ON

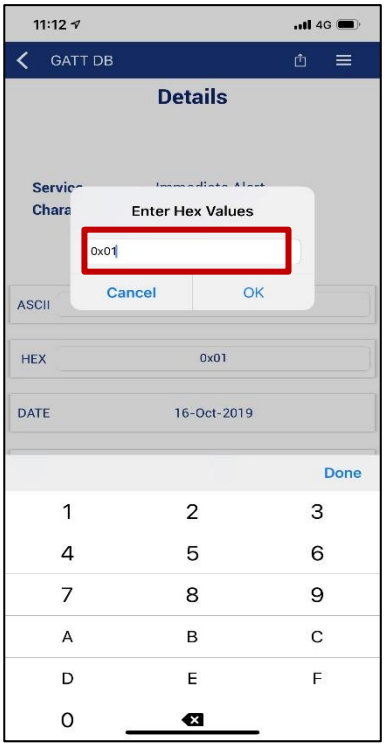


Figure 4 Alert Level Hex Value Input

4.2 Test Using the CySmart Host Emulation Tool

1. Connect the CySmart BLE dongle to your Windows PC. Wait for the driver installation to complete.

Note: If the BLE dongle shows up as a mass storage device, hold the Reset button for five seconds to change modes.

2. Launch the CySmart Host Emulation Tool by right-clicking on the BLE Component in PSoC Creator and selecting **Launch CySmart**. It automatically detects the BLE dongle. Click **Refresh** if the BLE dongle does not appear in the **Select BLE Dongle Target** pop-up window.

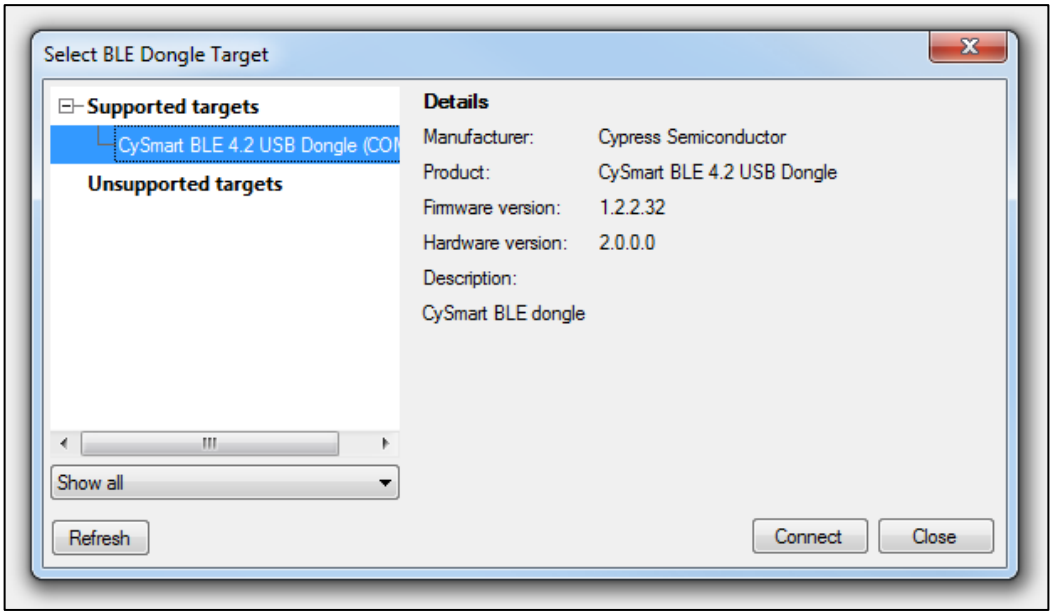


Figure 5 CySmart Dongle Selection

3. Reset the development kit to start advertising by pressing the Reset button (SW2) or by pressing the User button (SW2).
4. Click **Start Scan** to discover available devices.

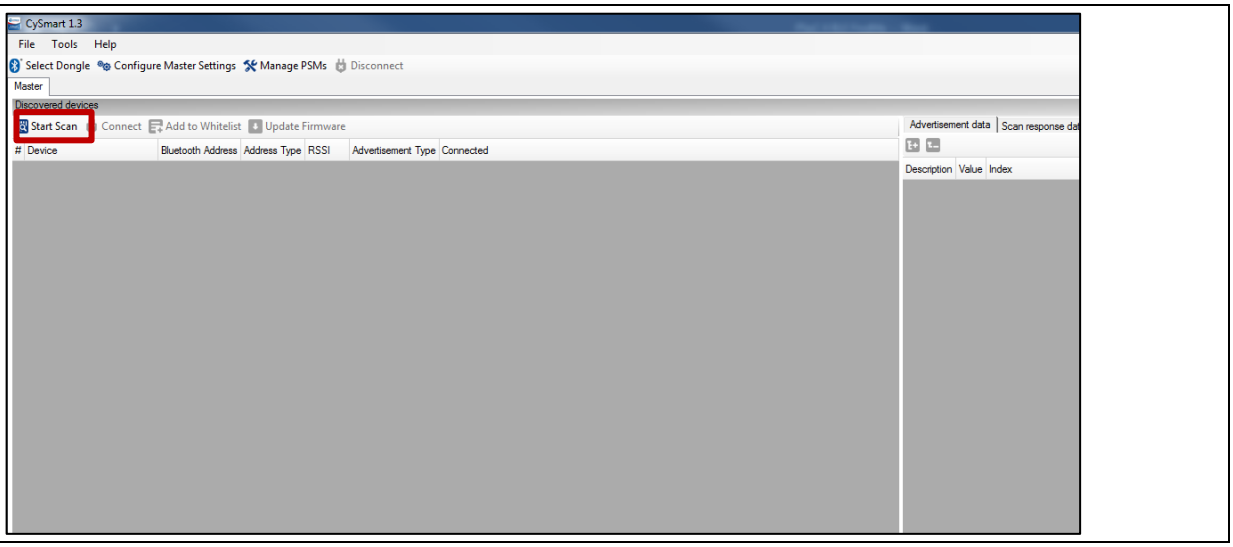


Figure 6 CySmart Scan for Devices

5. Select **Find me Target** from the list of available devices, and connect to it.

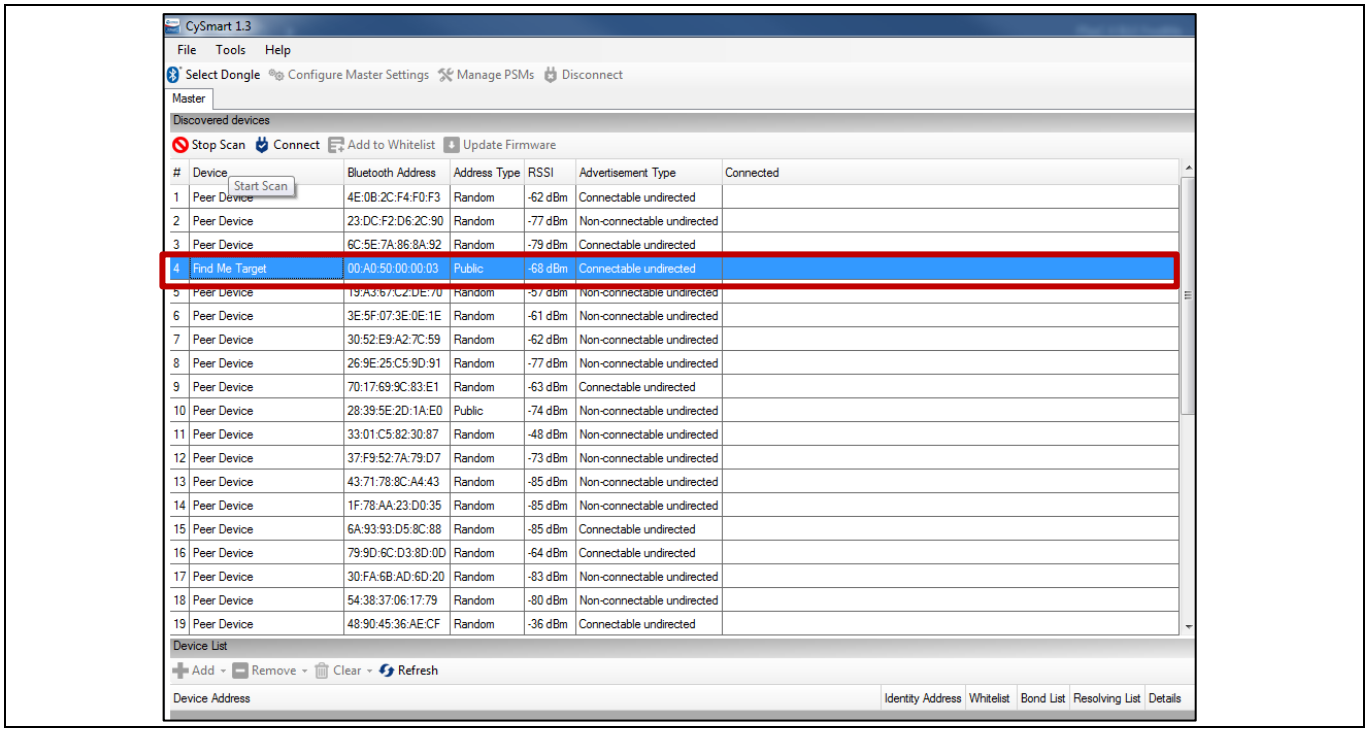


Figure 7 CySmart Device Target Selection

6. Switch to the **Find Me Target** device tab, and click **Discover all Attributes**.

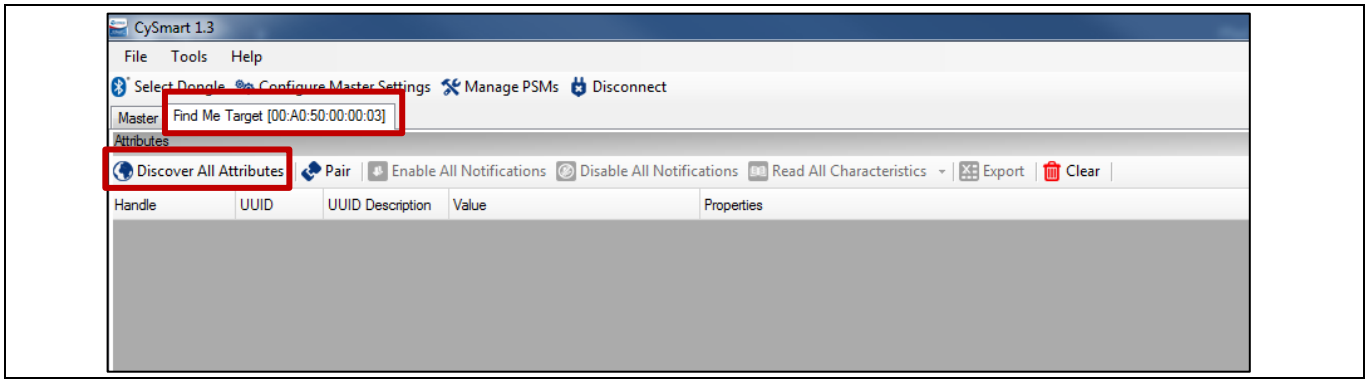


Figure 8 Discover Device Attributes

7. Select the **Alert Level characteristic** value.

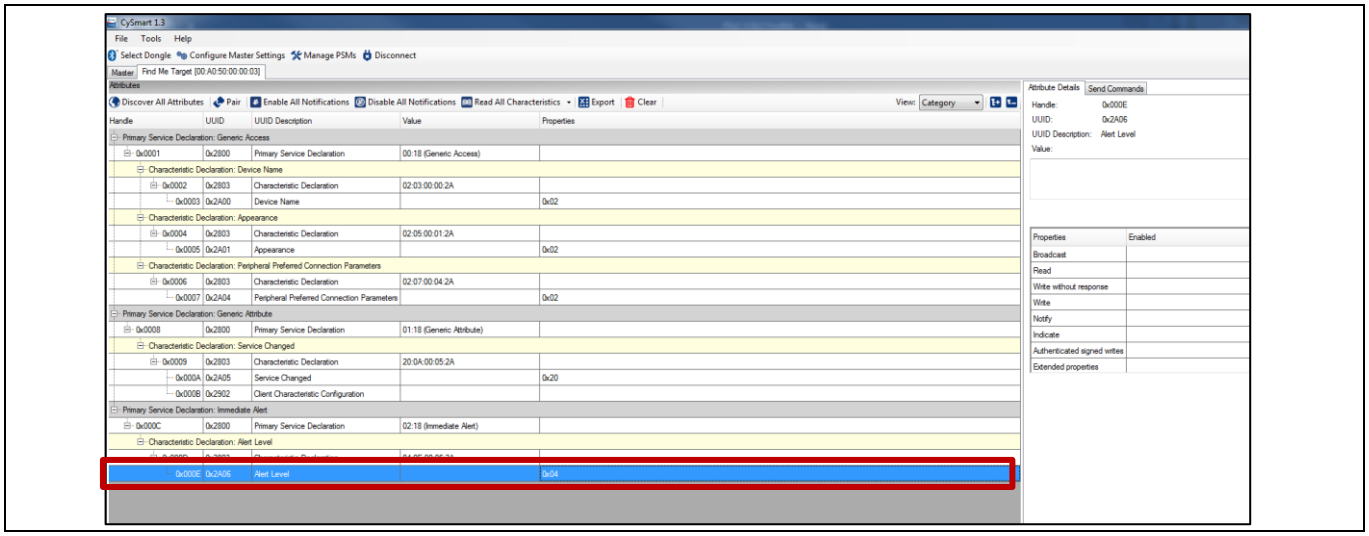


Figure 9 Alert Level Characteristic Selection

8. Write “2” (high alert), and click **Write Value Without Response**.

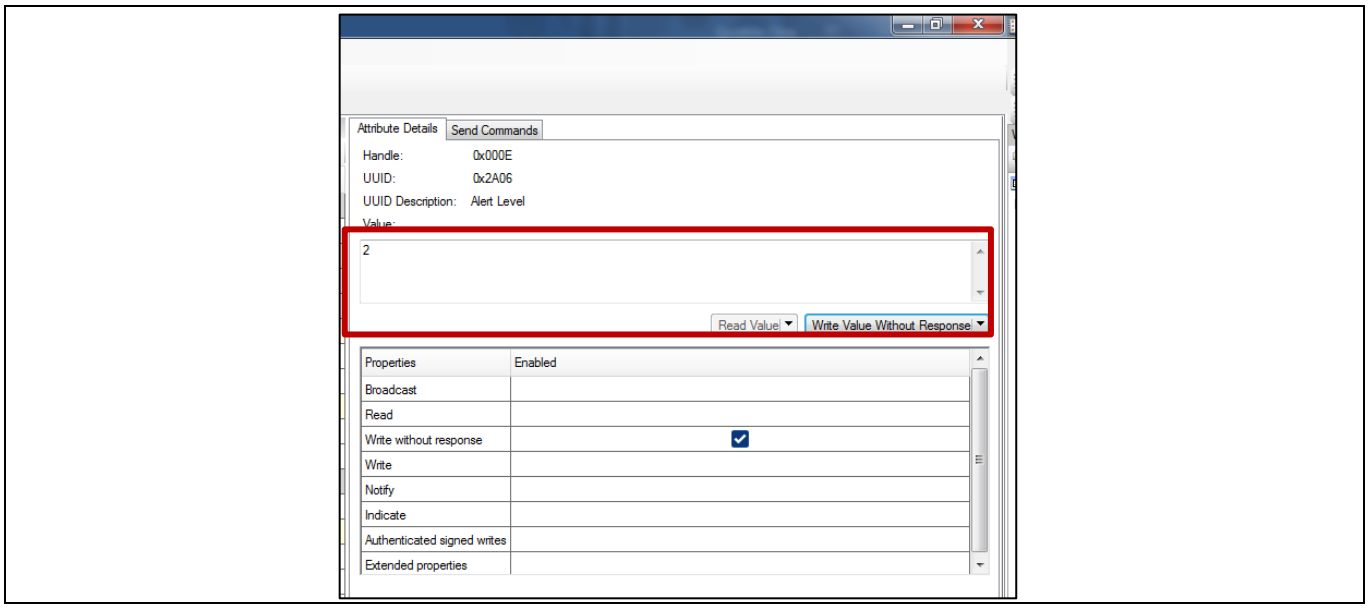


Figure 10 Writing Alert Level

9. Observe the blue LED turn ON, but not blinking.

5 Design and Implementation

The Bluetooth Low Energy (BLE) PSoC Creator Component provides a comprehensive GUI-based configuration window to facilitate designing applications requiring BLE connectivity. The Component incorporates a Bluetooth Core Specification v4.2- compliant protocol stack. The PSoC 4 BLE Find Me code example configures the BLE Component based on the specification laid out in the **Component and Settings** section.

5.1 BLE

This example uses the BLE_v3.63 Component. The component is configured with the Find Me profile. See the **Profiles** tab of the BLE Component Configurator for specific services and characteristics of the services of the Find Me profile. The 'Find Me Locator' (the BLE Central device) is a BLE GATT Client. The 'Find Me Target' (the Peripheral device) is a BLE GATT Server with the Immediate Alert Service.

Figure 11 shows the functional flow of the BLE Component connection.

The following functions are performed:

1. System initialization.
2. A callback function is set up for Immediate Alert Service.
3. Waiting for Bluetooth connection; the LED color set to green.
   If a connection is not made within 30 seconds, the LED color turns to red. Press the User button (SW2) after 30 seconds to restart advertisement for Bluetooth connection.
4. Connection made; LED turned OFF.
5. The BLE Central device changes the value of the alert state.
6. A switch statement selects a case depending on the alertLevel variable. For NO_ALERT, MILD_ALERT and HIGH_ALERT, the LED state is OFF, blinking, and ON respectively.
7. Enter Deep Sleep mode after BLE addressing times out when no device has connected.
8. When the User button (SW2) pressed, isr_wakeup is executed to wake up the device and restart BLE advertisement. The application flow goes back to Step 3.

The top-level design of the PSoC Creator project is shown in **Figure 12**. A detailed functional flow of the BLE Component connection is shown in **Figure 11**.

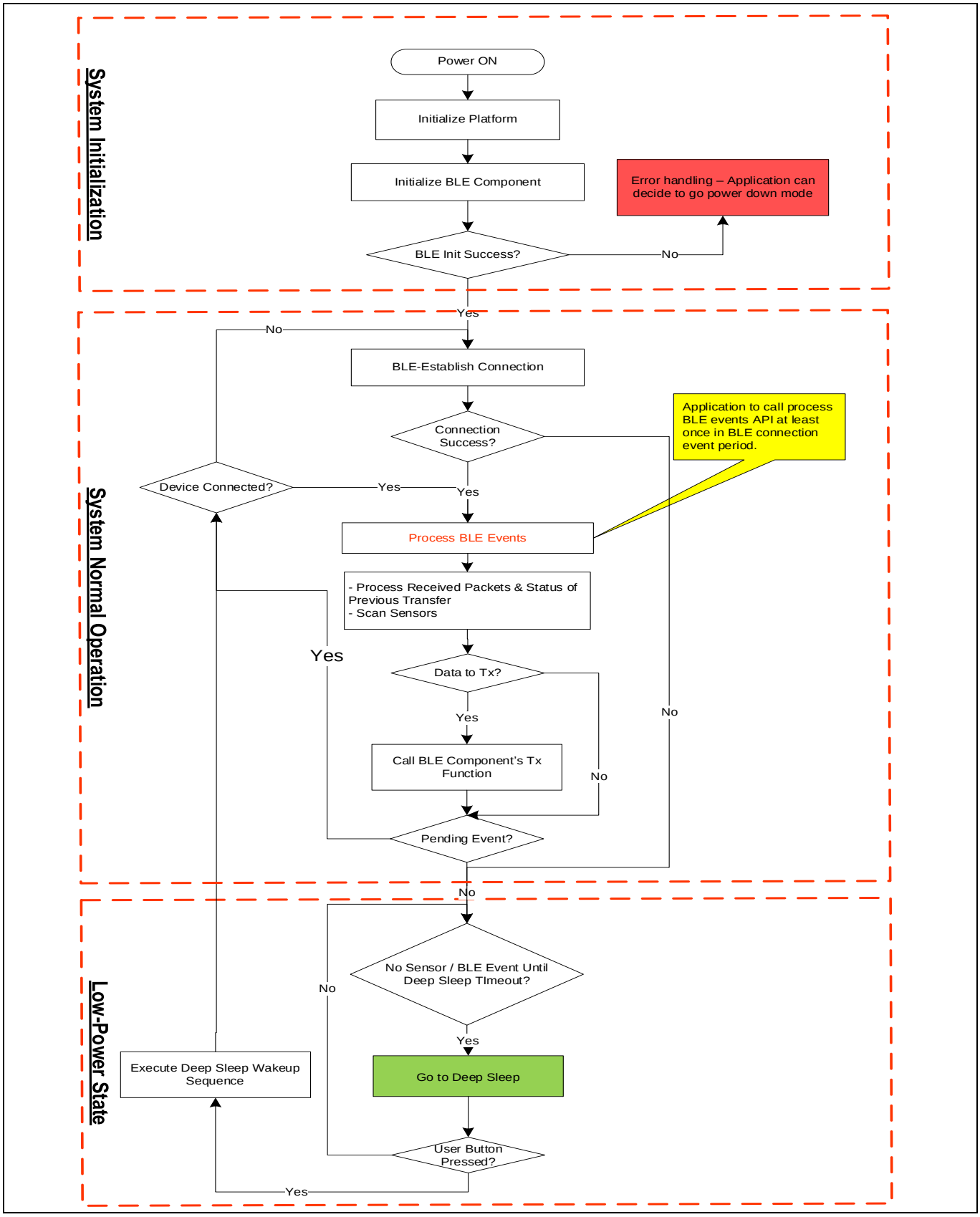


Figure 11 Bluetooth Functional Flow

Design and Implementation

The BLE Find Me profile defines what happens when the locating Central device transmits a change in the alert level. The Find Me locator performs service discovery using the 'GATT Discover All Primary Services' procedure. The BLE Service Characteristic discovery is done by the 'Discover All Characteristics of a Service' procedure. When the Find Me Locator wants to cause an alert on the Find Me Target, it writes an alert level in the Alert Level Characteristic of the Immediate Alert Service. When the Find Me Target receives an alert level, it displays the level using the blue LED: OFF for no alert, blinking for mild alert, and ON for high alert.

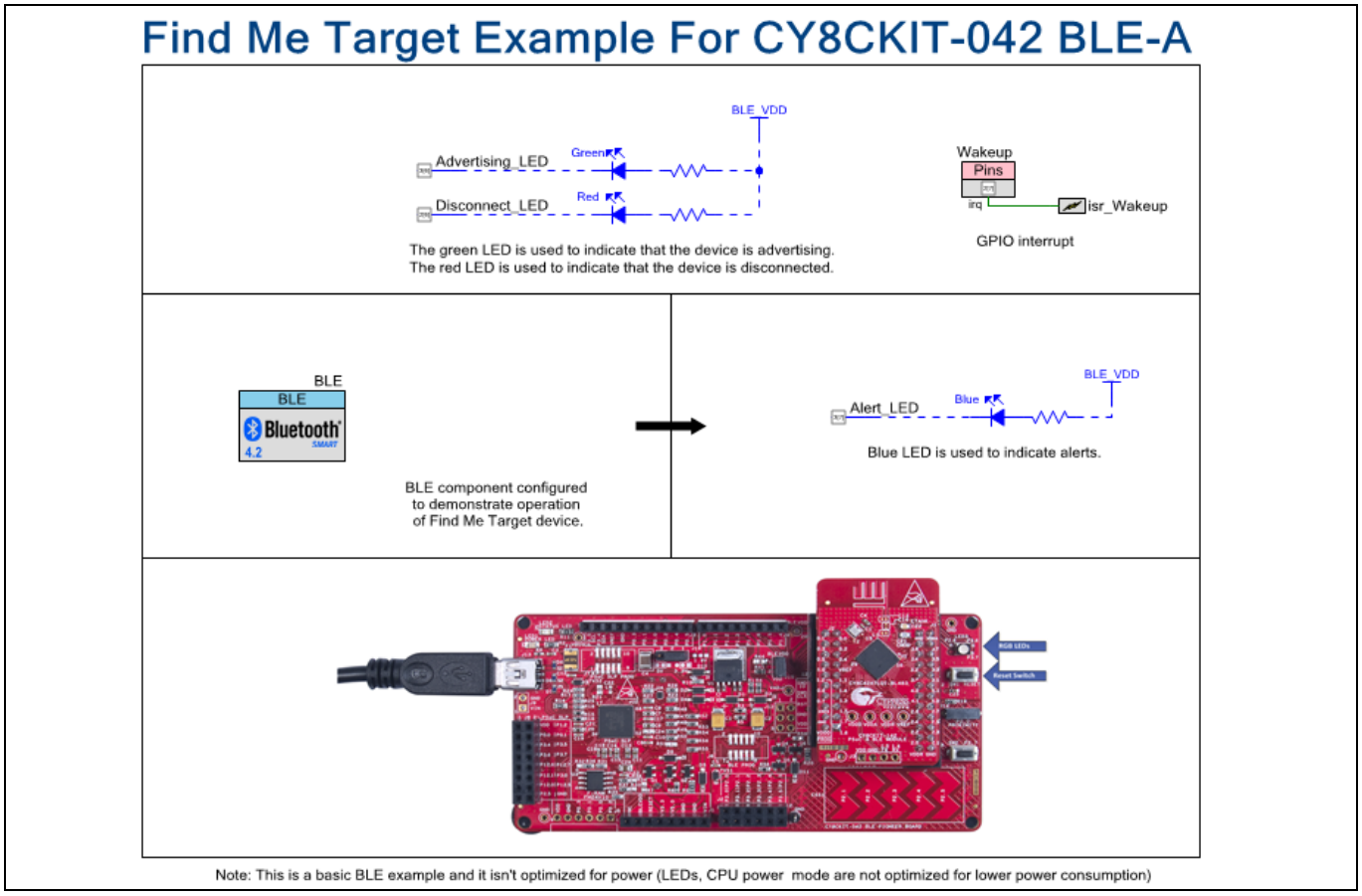


Figure 12 Top-level Design

6 Components and Settings

The following table lists the PSoC Creator Components used in this example, how they are used in the design, and the non-default settings required so they function as intended.

Table 1 PSoC Creator Components

Component	Instance Name	Purpose	Non-default Settings
Bluetooth Low Energy (BLE)	BLE	Implement BLE communication	General: **Figure 13** **GAP Settings**- General: **Figure 14** Advertisement Settings: **Figure 15** Advertisement Packet: **Figure 16** Scan response packet: **Figure 17** Security: **Figure 18**
Interrupt	Isr_wakeup	Interrupt that wakes up the device from deep sleep	**InterruptType:** Derived
Digital Input Pin	Wakeup	Digital Input Pin connected to “User” button (SW2). This triggers an interrupt	**Digital Output HW connection:** Off **Interrupt:** Falling Edge
Digital Output Pins	Advertising_LED	Provide visual feedback	**Digital Output HW connection:** Off **Initial drive state:** High
	Disconnect_LED		
	Alert_LED		

For information on the hardware resources used by a Component, see the Component datasheet.

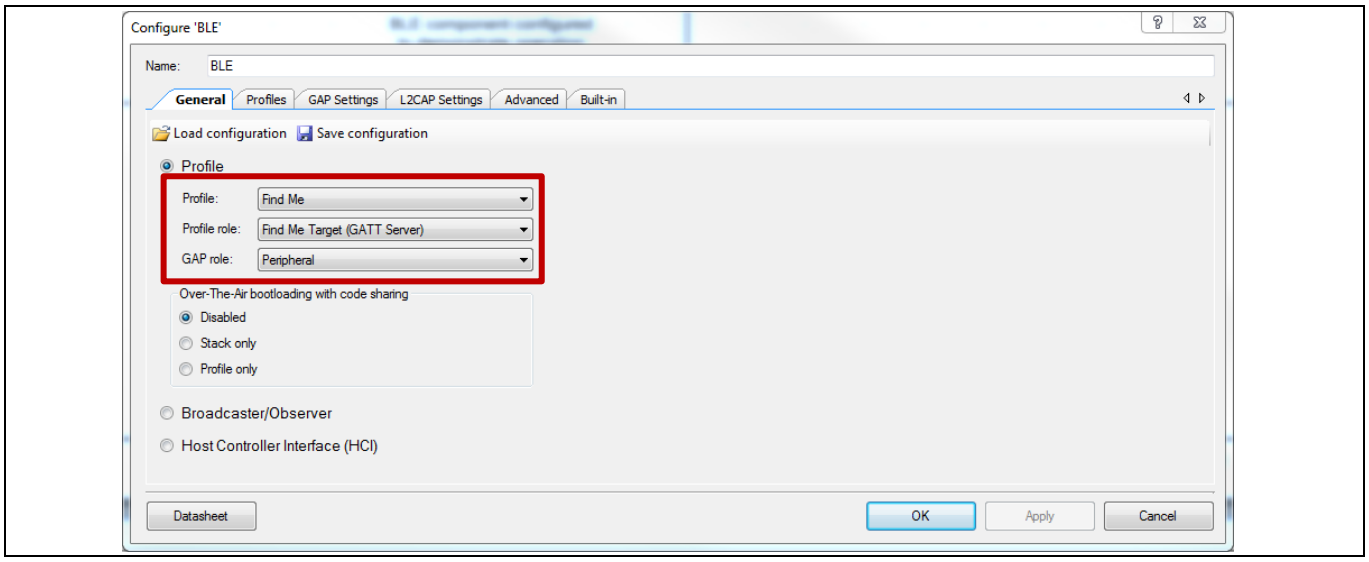


Figure 13 General Settings

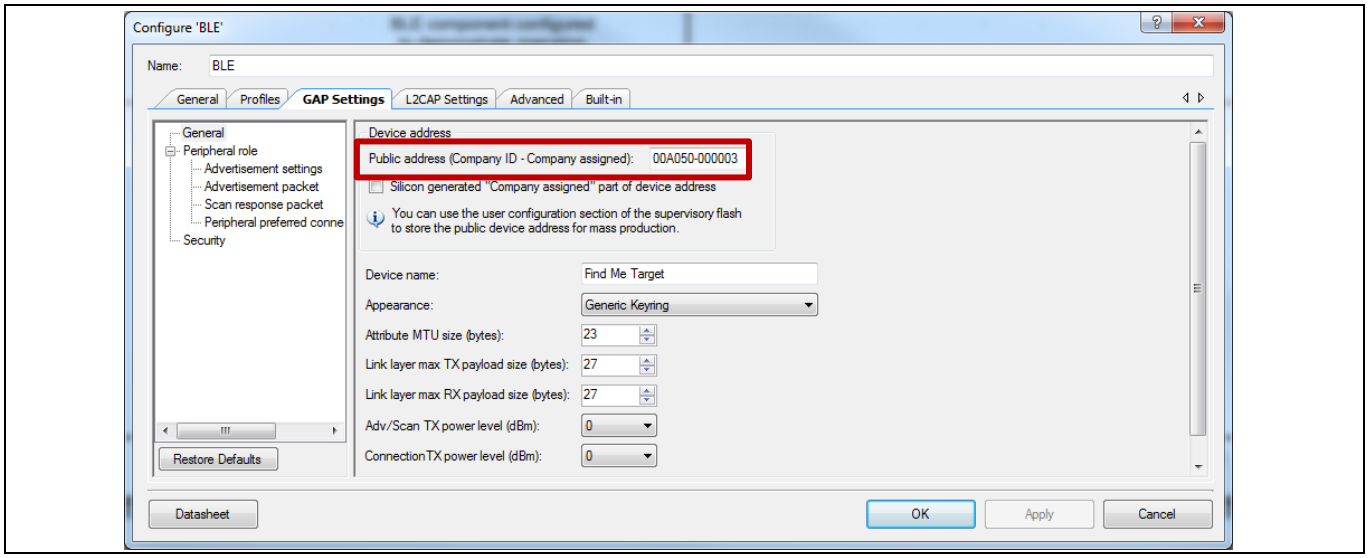


Figure 14 GAP Settings - General

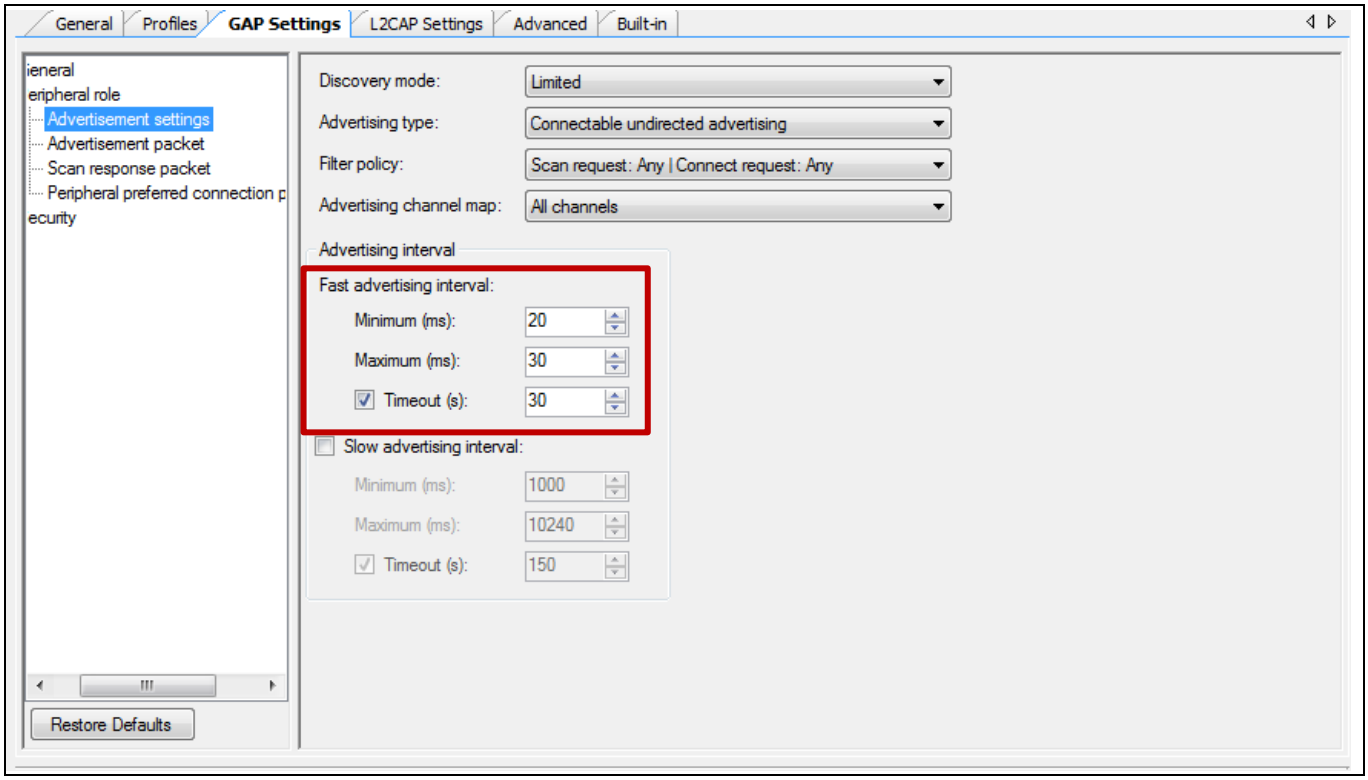


Figure 15 GAP Settings - Advertisement Settings

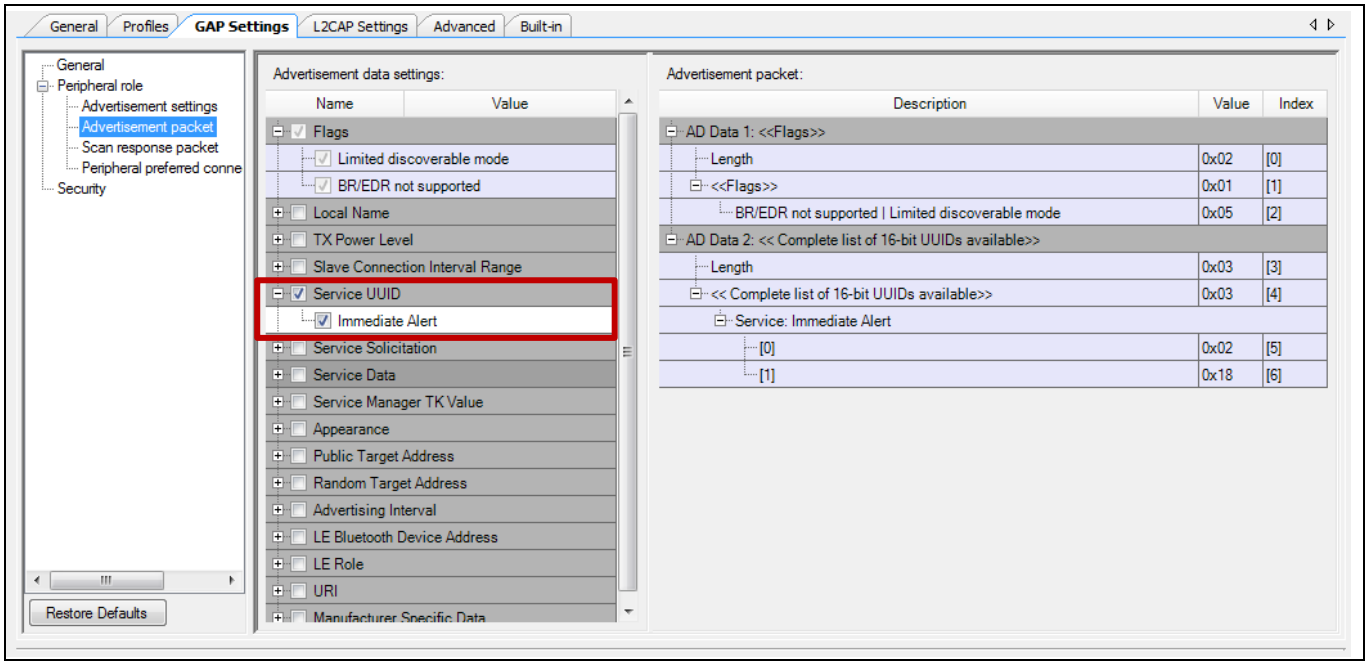


Figure 16 GAP Settings – Advertisement Packets

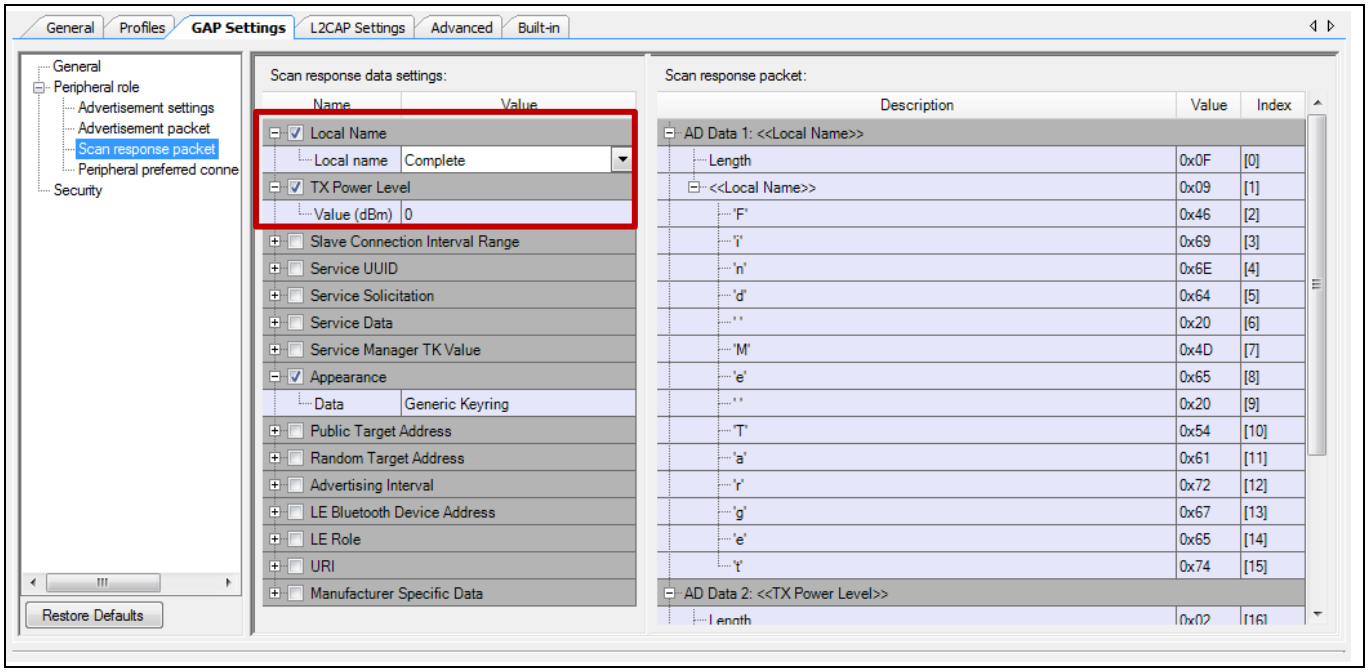


Figure 17 GAP Settings – Scan Response Packets

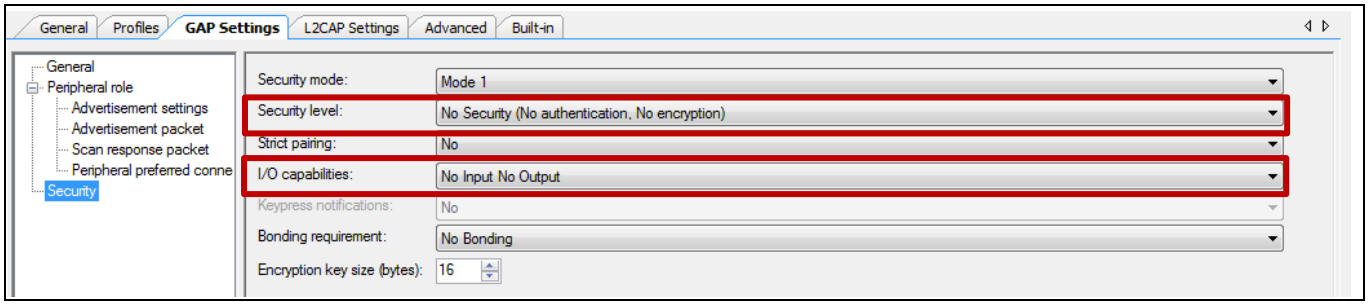


Figure 18 GAP Settings - Security

7 Reusing This Example

This example can be ported to various PSoC 4 devices, kits, or both. Before porting this example, note the following:

- Not all PSoC 4 devices have BLE.
- Pinouts change from kit to kit. Some pins may need to be moved. See the **Pin Layout** tab in PSoC Creator. To port the code to a new device, in PSoC Creator, select **Project** > **Device Selector**, and change the target device.

In some cases, a resource used by a code example (for example, a BLE block) is not supported on another device. In that case the example will not work. If you build the code targeted at such a device, you will get errors. See the device datasheet for information on what a particular device supports.

8 References

For a comprehensive list of PSoC 3, PSoC 4, and PSoC 5LP resources, see **KBA86521** in the Cypress community.

Application Notes

[1] **AN91267** – Getting Started with PSoC 4 BLE: Describes PSoC 4 MCU with BLE Connectivity devices and how to build your first PSoC Creator project

[2] **AN79953** – Getting Started with PSoC 4: Describes getting started with PsoC4 MCU

Related Code Examples

[3] **CE224734** – BLE HTTP Proxy Code Examples with PSoC® 4 BLE: Demonstrates the HTTP Proxy Client and Server operation of the BLE PSoC Creator Component

[4] **CE224734** – Bluetooth Low Energy (BLE) Object Transfer Client / Server: Demonstrates how to create an object transfer and object managing system using BLE PSoC Creator Component

PSoC Creator Component Datasheets

[5] **Pins:** Supports connection of hardware resources to physical pins

[6] **Timer Counter (TCPWM):** Supports fixed-function Timer/Counter implementation

[7] **Clock:** Supports local clock generation

[8] **Interrupt:** Supports generating interrupts from hardware signals

[9] **BLE:** Supports PSoC Bluetooth Low Energy

Device Documentation

[10] **PSoC 4 Technical Reference Manuals:** PSoC 4 Datasheets

[11] **PSoC 4 Datasheets:** PSoC 4 Datasheets

Development Kit Documentation

[12] **CY8CKIT-042-BLE-A Bluetooth® Low Energy 4.2 Compliant Pioneer Kit**

Tool Documentation

[13] **PSoC Creator:** Look in the downloads tab for Quick Start and User Guides

9 Revision History

Revision	Date	Description of Change
**	2019-12-16	New code example
*A	2021-1-20	Sunset review, made minor changes

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

Edition 2020-12-22

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
002-28696 Rev. *A

IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.