Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

CE212736 - PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity - Find Me

Objective

This code example demonstrates the implementation of a simple BLE Immediate Alert Service (IAS)-based Find Me profile using PSoC® 6 MCU with BLE Connectivity.

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 SIG. The design uses the RGB LED on the CY8CKIT-062-BLE PSoC 6 BLE 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 Peripheral device (the Pioneer kit) is advertising or disconnected.

The USB-BLE dongle provided with the CY8CKIT-062-BLE Pioneer kit or an iOS/Android mobile device can act as the BLE Central device, which locates the Peripheral device.

Requirements

Tool: PSoC Creator™ 4.2; Peripheral Driver Library (PDL) 3.0.1

Programming Language: C (Arm® GCC 5.4.1 and Arm MDK 5.22)

Associated Parts: All PSoC 6 MCU dual-core parts with BLE connectivity

Related Hardware: CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit

Hardware Setup

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

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 behavior with any of the two options, but the CySmart app is simpler.

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

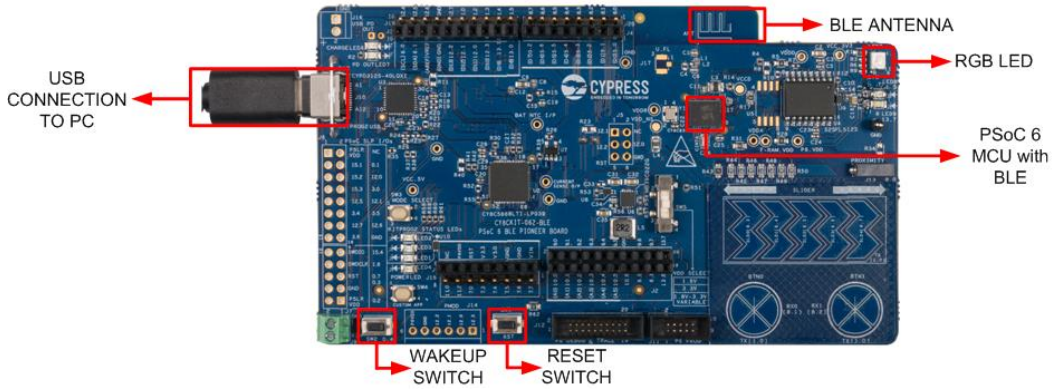
iOS

Android

Operation

1. Plug the CY8CKIT-062-BLE kit board into your computer's USB port.

Figure 1. CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit Baseboard



2. Build the project and program it into the PSoC 6 MCU device. Choose **Debug** > **Program**. For more information on device programming, see PSoC Creator Help. Flash for both CPUs is programmed in a single program operation.
3. Observe the greed LED turn ON after the device starts advertisement.
4. To test using the CySmart mobile app:
    a. Turn ON Bluetooth on your Android or iOS device.
    b. Launch the CySmart app.
    c. Press the reset switch on the Pioneer Kit to start BLE advertisements from your design. The advertisement LED (green LED) turns ON to indicate that BLE advertisement has started.
    d. 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 to establish a BLE connection. Once the connection is established, the green LED turns OFF.
    e. Select the 'Find Me' Profile from the carousel view.
    f. 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.

Figure 2 shows the steps for using the CySmart App on iOS. Figure 3 does the same for Android.

Figure 2. Testing with the CySmart App on iOS

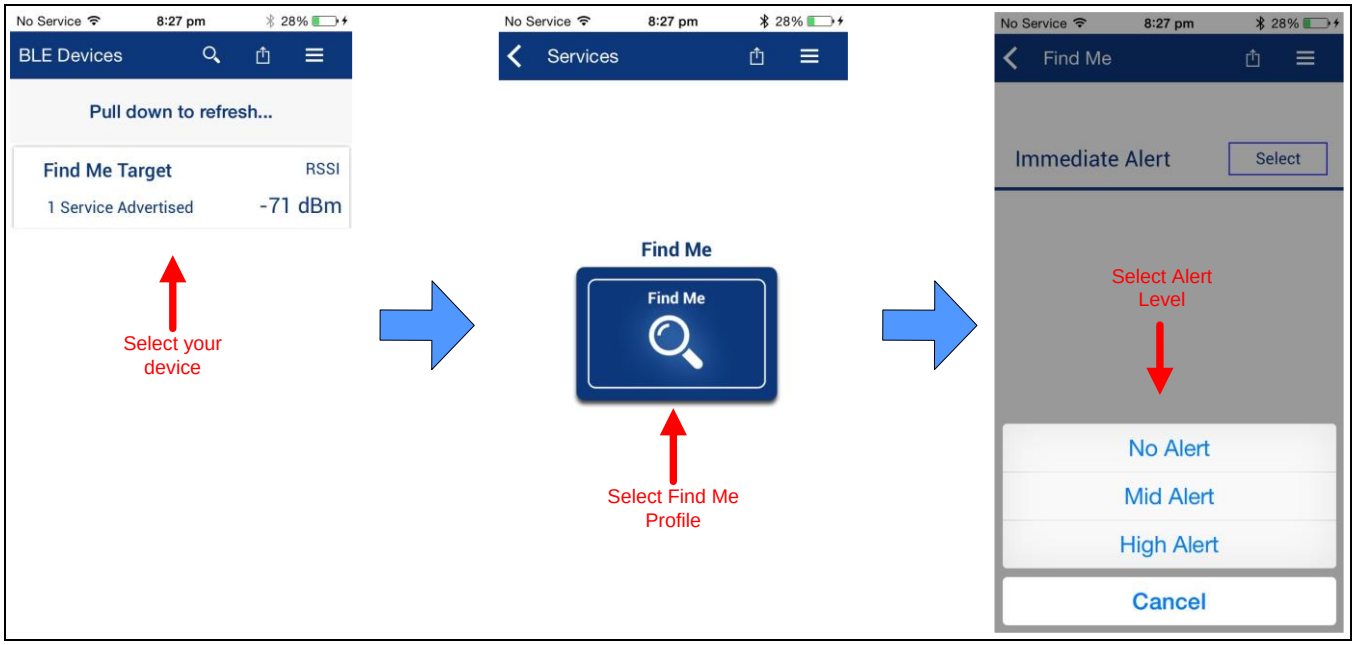
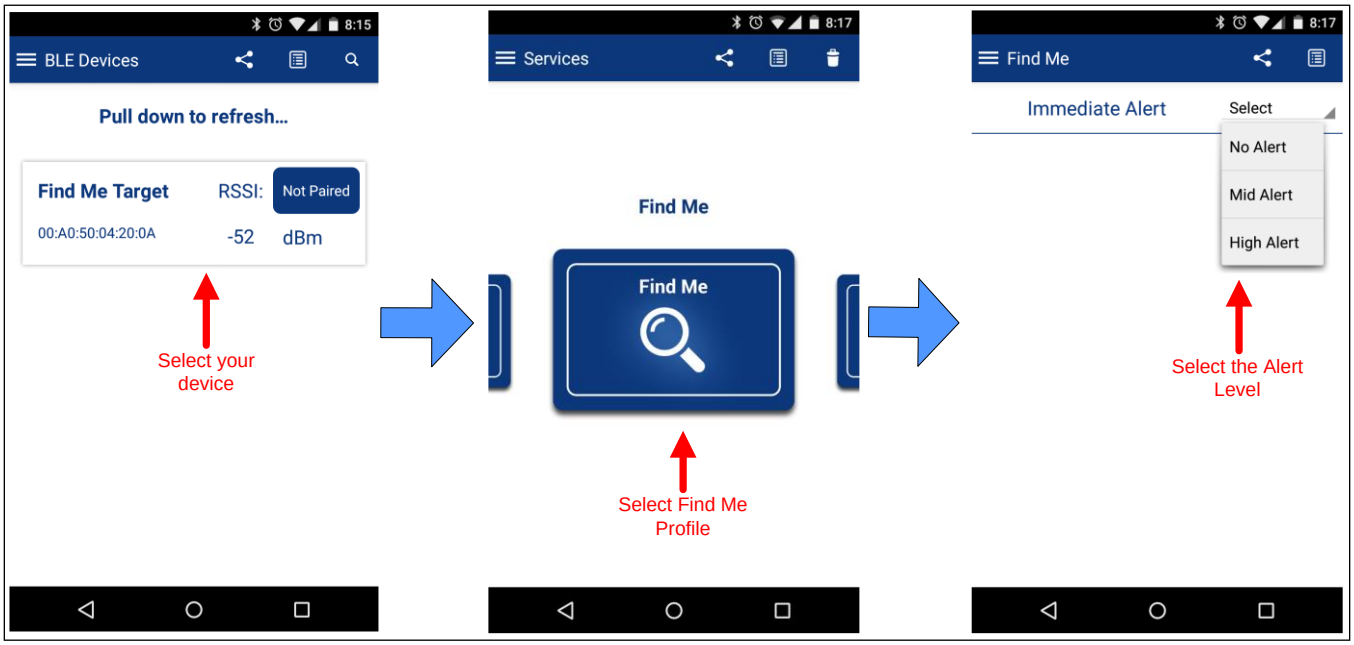
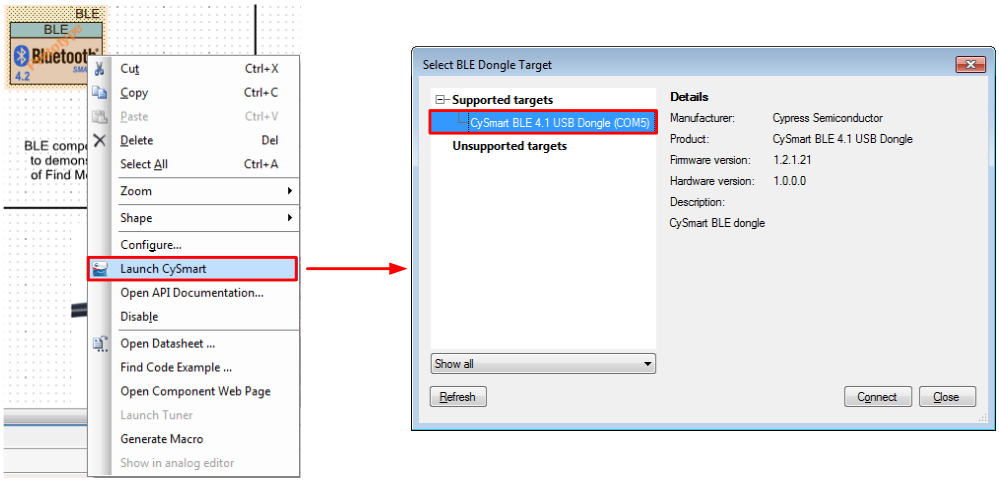


Figure 3. Testing with the CySmart App on Android



5. To test using the CySmart Host Emulation Tool:
    a. Connect the BLE Dongle to your Windows PC. Wait for the driver installation to complete.
    b. Launch the CySmart Host Emulation Tool by right-clicking on the BLE Component 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. Click **Connect,** as shown in Figure 4.

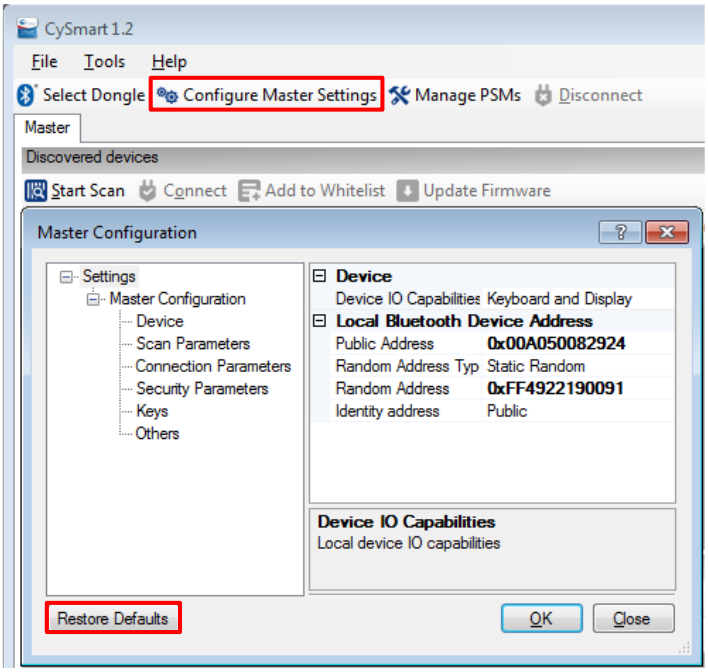
Figure 4. CySmart BLE Dongle Selection



Note: If the dongle firmware is outdated, you will be alerted. You must upgrade the firmware before you can complete this step. Follow the instructions in the window to update the dongle firmware.

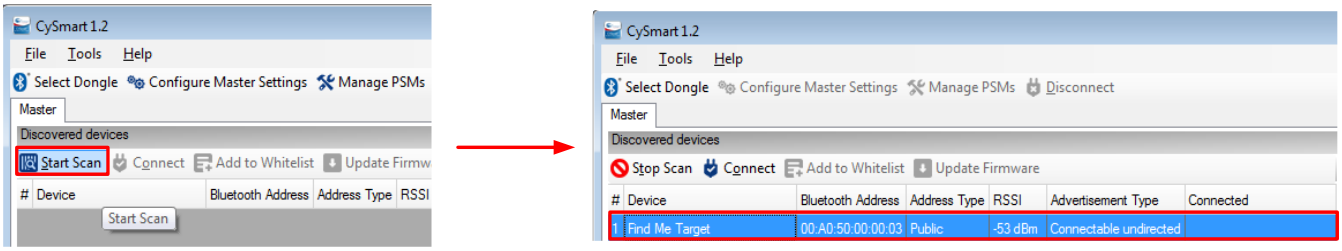
    c. Select **Configure Master Settings** and then click **Restore Defaults**, as shown in Figure 5. Then click **OK**.

Figure 5. CySmart Master Settings Configuration



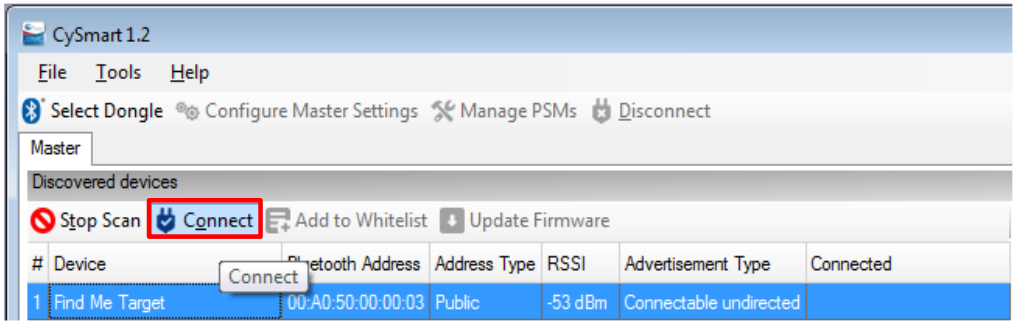
    d. Press the reset switch on the Pioneer Kit to start BLE advertisements from your design.
    e. On the CySmart Host Emulation Tool, click **Start Scan**. Your device name (configured as Find Me Target) should appear in the Discovered devices list, as shown in Figure 6.

Figure 6. CySmart Device Discovery



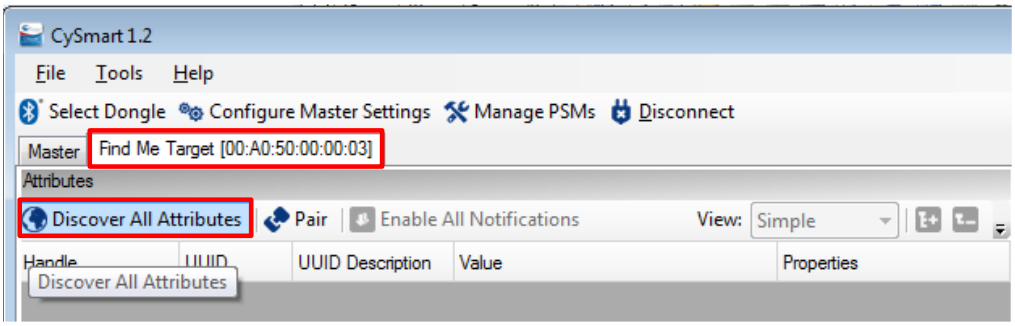
f. Select your device and click **Connect** to establish a BLE connection between the CySmart Host Emulation Tool and your device, as shown in Figure 7.

Figure 7. CySmart Device Connection



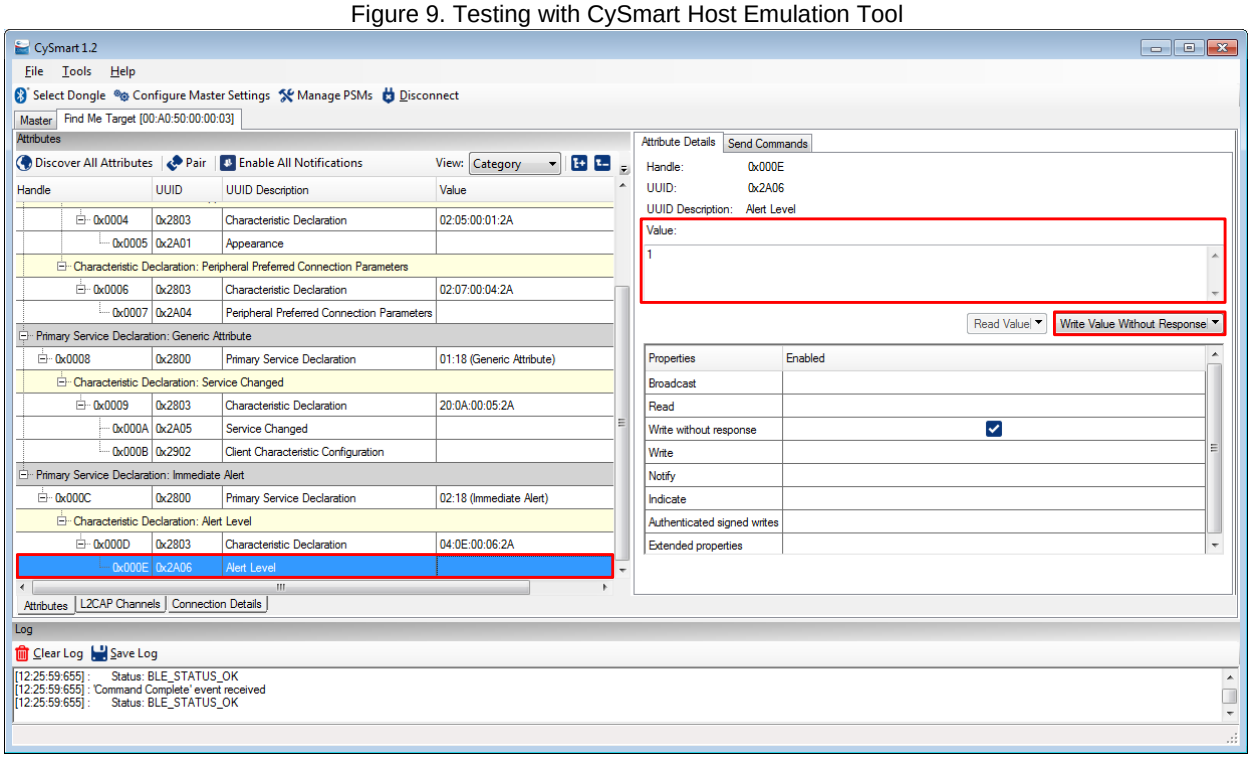
g. Once connected, switch to the **Find Me Target** device tab and discover all the Attributes on your design from the CySmart Host Emulation Tool, as shown in Figure 8.

Figure 8. CySmart Attribute Discovery



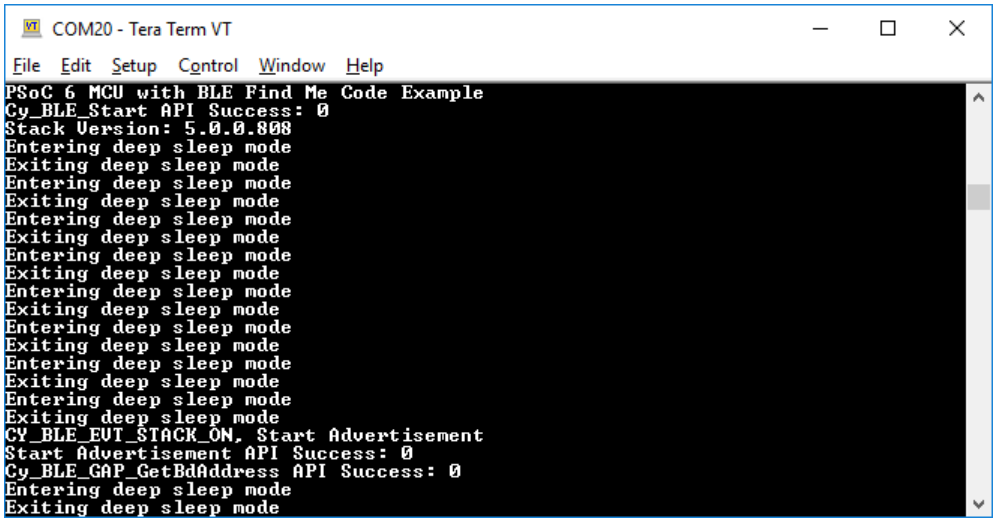
h. Scroll down the **Attributes** window and locate the **Immediate Alert** Service fields. Write a value of 0 – no alert, 1 – mid alert, or 2 – high alert to the **Alert Level** Characteristic under the **Immediate Alert Service**, as Figure 9 shows. Observe that the state of the LED on your device changes per your Alert Level Characteristic configuration.

Figure 9. Testing with CySmart Host Emulation Tool



6. Use the UART debug port to view verbose messages:
    a. The code example ships with the debug port disabled. To enable it, set the macro `DEBUG_UART_ENABLED` in *debug.h* to `ENABLED` and rebuild the code.
    b. Use your favorite serial terminal application and connect to the **KitProg2 USB-UART** COM port. Configure the application to access the COM port at 115200 bps baud rate.
    c. Program the board. The debug messages will appear in the terminal window as shown in Figure 10.

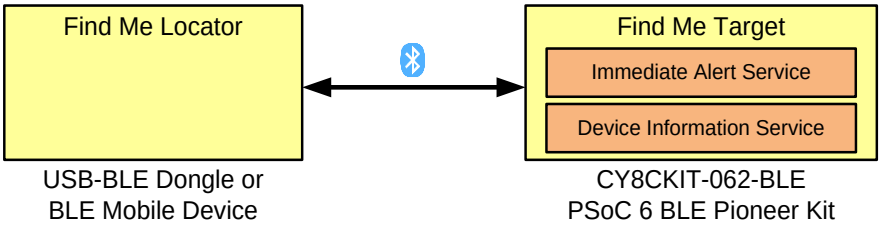
Figure 10. Debug Messages on COM Port



Design and Implementation

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 and an additional Device Information Service implemented, as Figure 11 shows.

Figure 11. Find Me Service Relationship



The BLE Find Me profile defines what happens when the locating Central device broadcasts 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. BLE Find Me Target Schematic

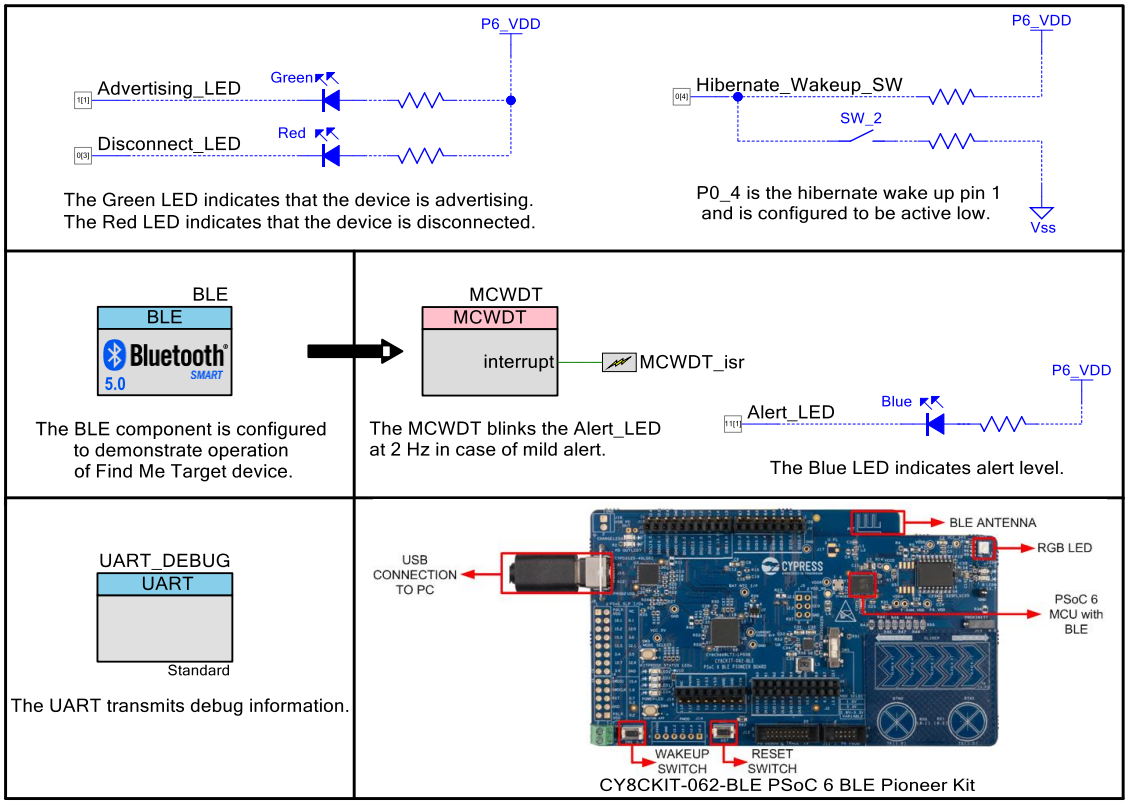


Figure 12 shows the top design schematic of the PSoC Creator project. The BLE interface is implemented on a PSoC 6 MCU with BLE Connectivity device using the BLE Component. The application runs primarily on the Arm Cortex®-M4 core. The Cortex-M0+ core executes the BLE controller firmware and is responsible for maintaining the BLE connection.

Refer to AN210781 – Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity to understand the design of firmware for this code example.

The device enters low-power Deep Sleep mode when BLE is idle. It wakes up automatically when there is activity on the BLE connection.

When BLE is disconnected, the device enters Hibernate mode. It wakes up when the reset switch or wakeup switch (SW2) is pressed and performs a complete reset sequence in firmware.

Components and Settings

Table 1 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	See below
Digital Input Pin	Hibernate_Wakeup_SW	Wake up device from hibernate	**Digital Input HW connection**: Off **Drive mode**: Resistive Pull Up **Initial drive state**: High
Digital Output Pin	Advertising_LED	Provide visual feedback	**Digital Output HW connection**: Off **Initial drive state**: High
	Alert_LED		
	Disconnect_LED		
MCWDT_PDL	MCWDT	Generate a tick every 250 msec	**Counter0** **Enable counter**: Checked **Match**: 7999 **Mode**: Interrupt **Clear on Match**: Clear on match
SysInt_PDL	MCWDT_isr	Wake up CM4 device based on MCWDT tick	**Deep Sleep Capable:** Checked
UART (SCB)	UART_DEBUG	Provide a serial interface for verbose messaging	Default settings only

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

Figure 13 through Figure 23 highlight the non-default settings for each component in this example. The settings for each Component are explained in detail in AN210781 – Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity.

Figure 13. BLE: Protocol Configuration

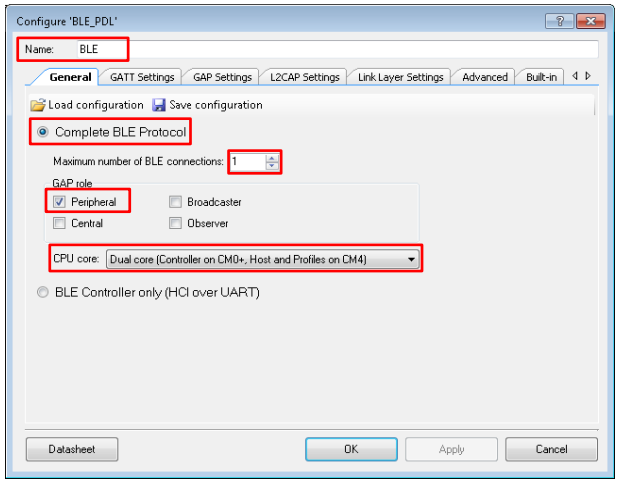


Figure 14. BLE: Adding Find Me Profile

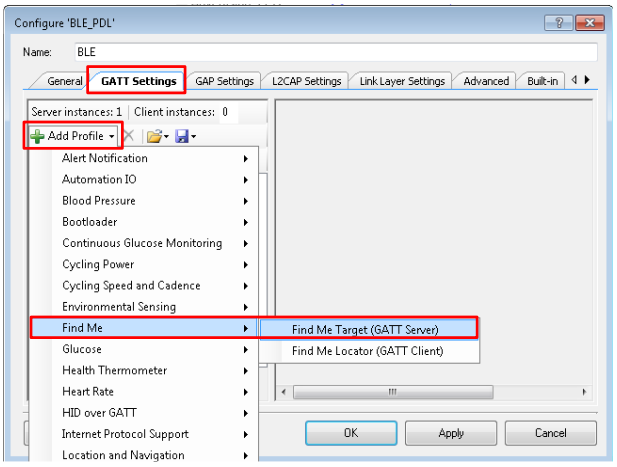


Figure 15. BLE: Device Configuration

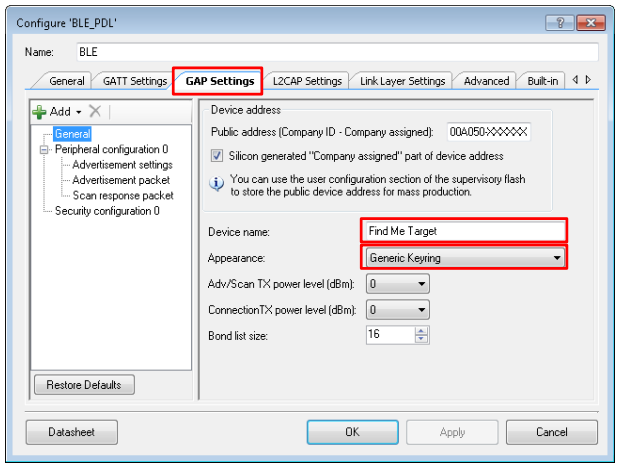


Figure 16. BLE: Advertisement Settings

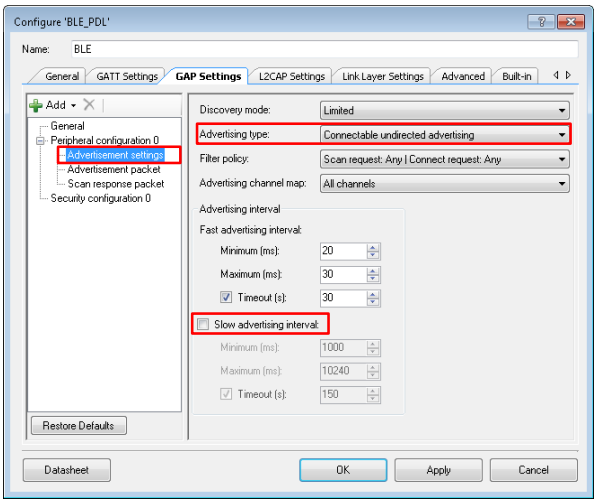


Figure 17. BLE: Advertisement Packet Settings

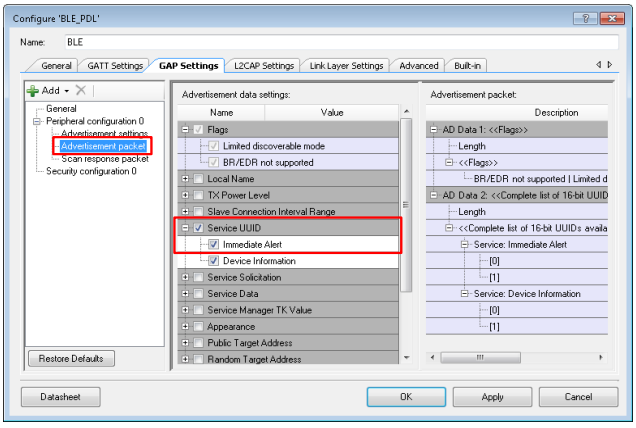


Figure 19. BLE: Response Packet Settings

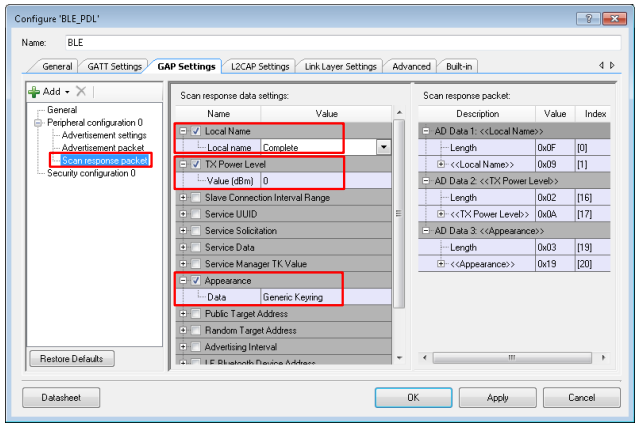


Figure 18. BLE: Security Configuration

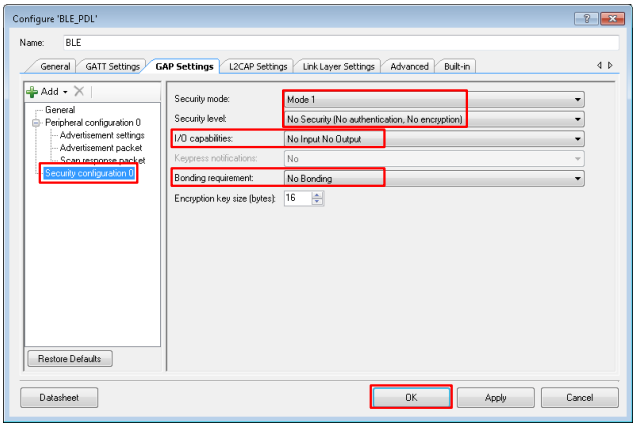


Figure 20. Advertising/Alert/Disconnect LED Settings

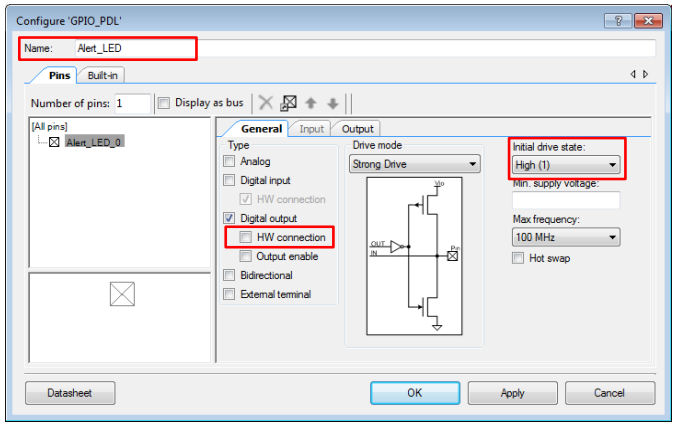


Figure 21. Hibernate Wakeup Pin Settings

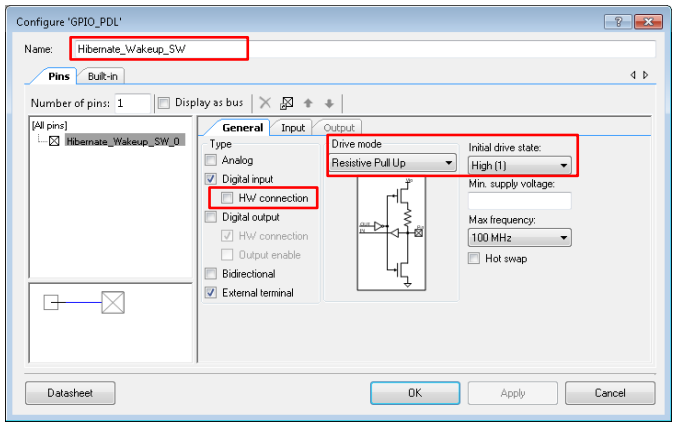


Figure 22. MCWDT ISR Settings

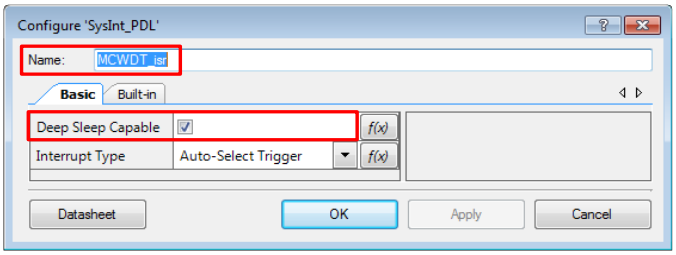


Figure 23. MCWDT Settings

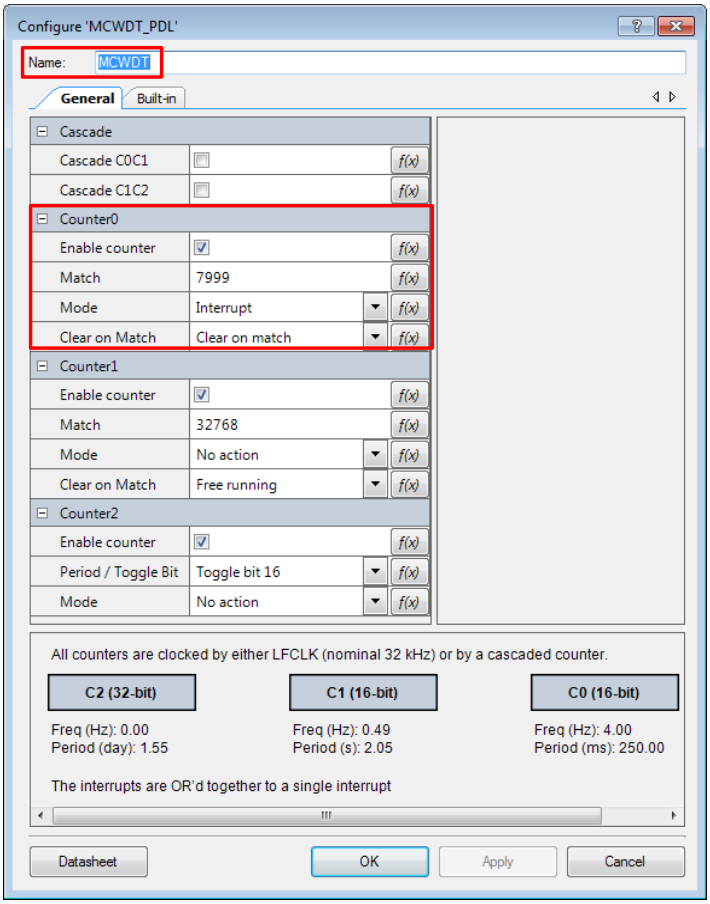


Figure 24 shows the system clock configuration.

Figure 24. Clock Configuration

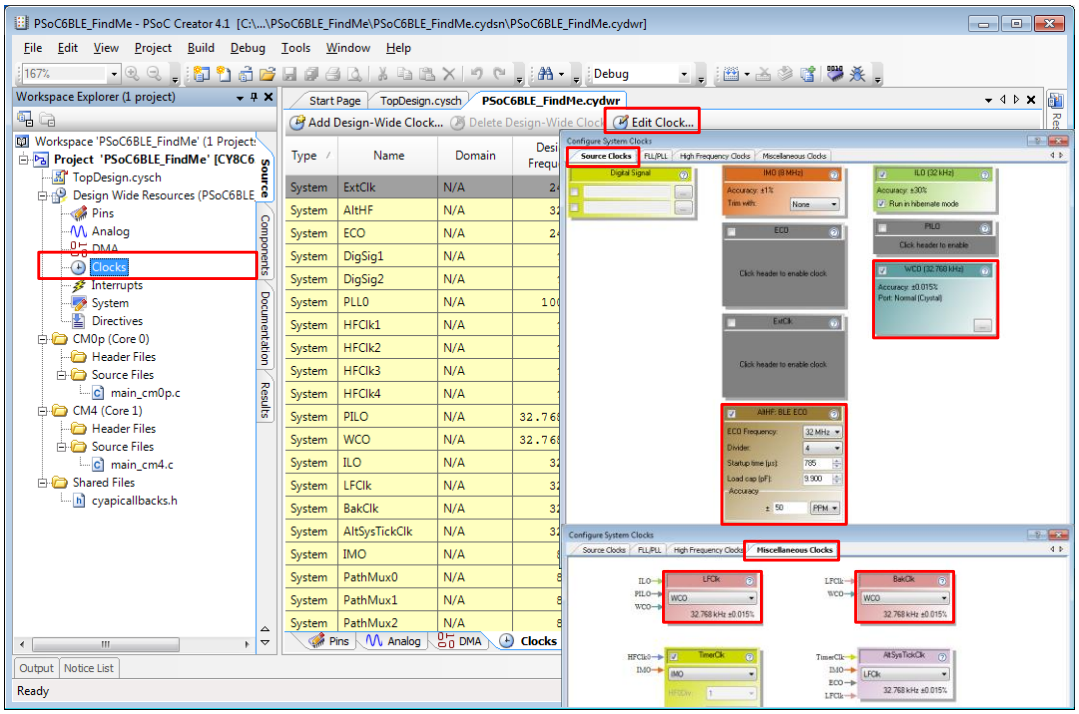
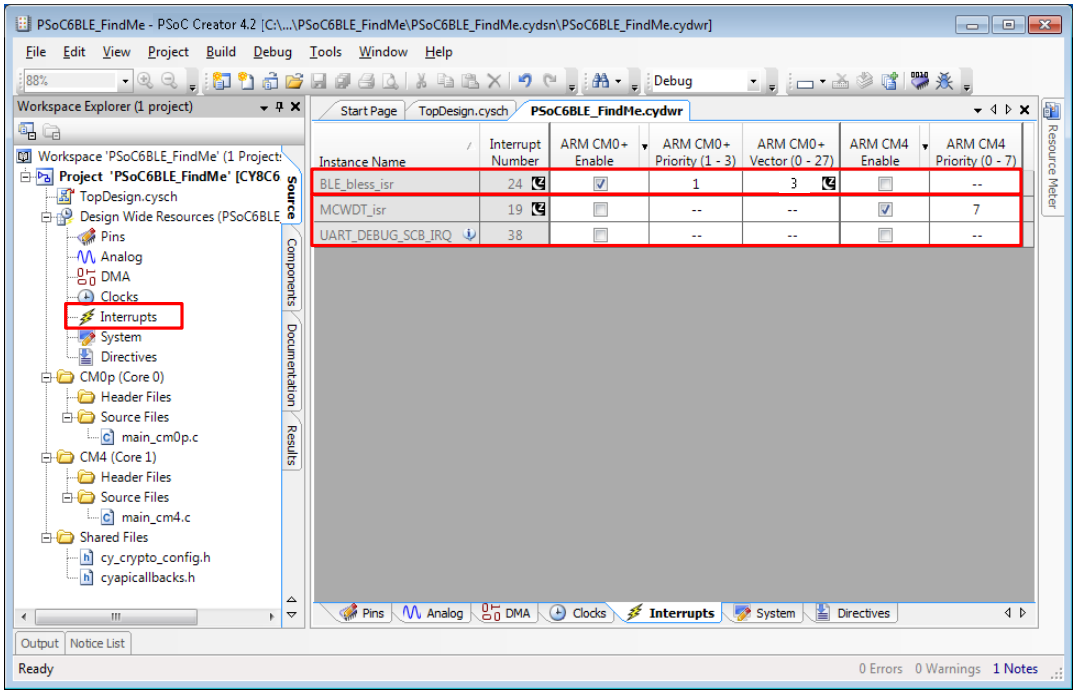


Figure 25 shows the system interrupt configuration.

Figure 25. Interrupt Configuration



Related Documents

Application Notes	
AN210781 – Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity	Describes PSoC 6 MCU with BLE Connectivity devices and how to build your first PSoC Creator project
AN215656 – PSoC 6 MCU: Dual-Core CPU system Design	Describes the dual-core CPU architecture in PSoC 6 MCU, and shows how to build a simple dual-core design
AN219434 – Importing PSoC Creator Code into an IDE for a PSoC 6 MCU Project	Describes how to import the code generated by PSoC Creator into your preferred IDE
PSoC Creator Component Datasheets	
Bluetooth Low Energy	Facilitates designing applications requiring BLE connectivity
Digital Input Pin	Supports connection of hardware resources to physical pins
Digital Output Pin	Supports connection of hardware resources to physical pins
MCWDT	Provides a timer
UART (SCB)	Provides asynchronous serial communications
ISR	Provides an interface for system interrupts
Device Documentation	
PSoC 6 MCU: PSoC 63 with BLE Datasheet	PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual
Development Kit Documentation	
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit	

PSoC Resources

Cypress provides a wealth of data at www.cypress.com to help you to select the right PSoC device for your design, and quickly and effectively integrate the device into your design. For a comprehensive list of resources, see KBA86521. The following is an abbreviated list of resources:

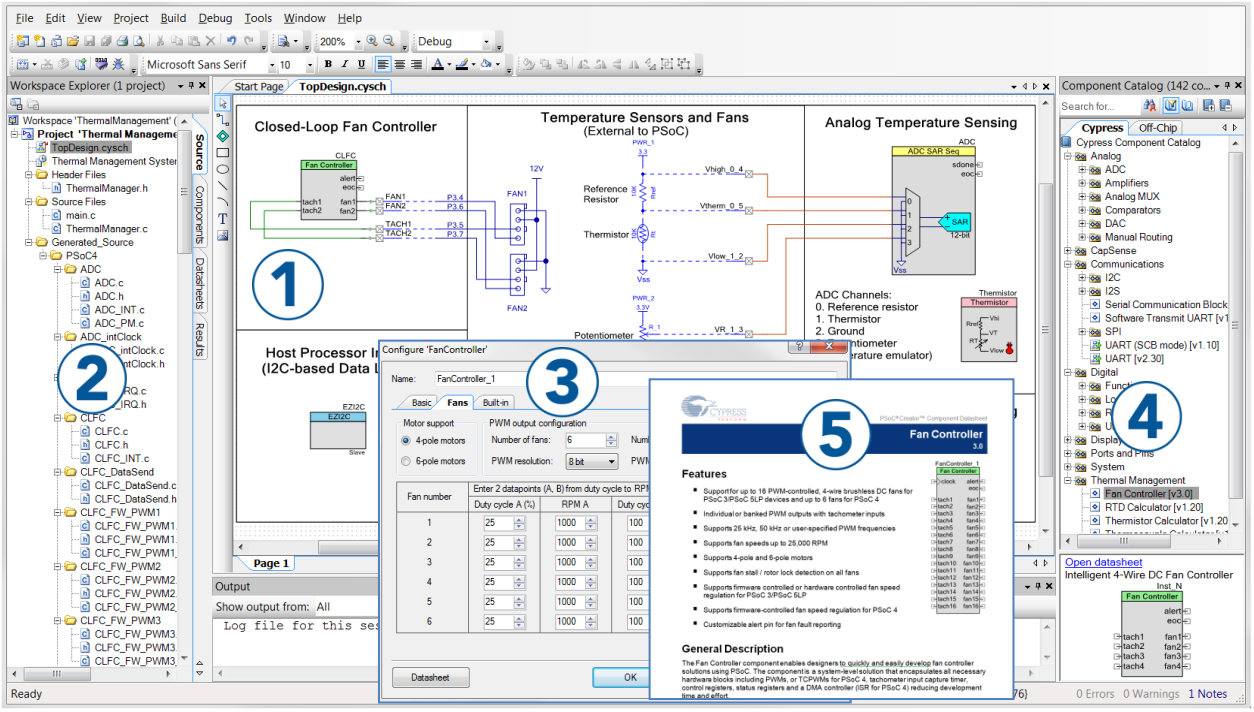
- **Overview:** PSoC Portfolio, PSoC Roadmap
- **Product Selectors:** PSoC 1, PSoC 3, PSoC 4, PSoC 5LP, or PSoC 6 MCU. In addition, PSoC Creator includes a device selection tool.
- **Datasheets:** Describe and provide electrical specifications for the PSoC device families.
- **CapSense Design Guides:** Learn how to design capacitive touch-sensing applications.
- **Application Notes:** Cover a broad range of topics, from basic to advanced level.
- **Code Examples**: for PSoC 3, PSoC 4, and PSoC 5LP; or for PSoC 6 MCU.
- **Technical Reference Manuals (TRM):** Provide detailed descriptions of the architecture and registers in each of the PSoC device families.
- **PSoC Training Videos**: These videos provide step-by-step instructions on getting started building complex designs with PSoC devices.
- **Development Kits:**
  - PSoC 6 BLE Pioneer Kit is a low-cost hardware platform that enables design and debug of the PSoC 63 series. It comes with an E-Ink display shield board.
  - CY8CKIT-042 and CY8CKIT-040, Pioneer kits, are easy-to-use and inexpensive development platforms. These kits include connectors for Arduino™ compatible shields and Digilent® Pmod™ daughter cards.
  - CY8CKIT-049 is a series of very low-cost prototyping platform for sampling PSoC 4 devices.
  - CY8CKIT-030 and CY8CKIT-050 are designed for analog performance. They enable you to evaluate, develop, and prototype high-precision analog, low-power, and low-voltage applications powered by PSoC 3 and PSoC 5LP, respectively.
  - CY8CKIT-001 is a common development platform for all PSoC family devices.
- The MiniProg3 device provides an interface for flash programming and debug.

PSoC Creator

PSoC Creator is a free Windows-based Integrated Design Environment (IDE). It enables concurrent hardware and firmware design of systems based on PSoC 3, PSoC 4, PSoC 5LP, and PSoC 6 MCU. See Figure 26 – with PSoC Creator, you can:

1. Drag and drop Components to build your hardware system design in the main design workspace
2. Codesign your application firmware with the PSoC hardware
3. Configure Components using configuration tools
4. Explore the library of 100+ Components
5. Review Component datasheets

Figure 26. PSoC Creator Features



Document History

Document Title: CE212736 – PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity - Find Me

Document Number: 002-12736

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5611559	SNVN	01/31/2017	Initial release
*A	5654163	SNVN	03/08/2017	Code updated to work with latest version of PSoC Creator Updated template
*B	5759463	SNVN	06/01/2017	Code updated to work with latest version of PSoC Creator
*C	5855193	SNVN	08/16/2017	Template and web links updated Code updated to work with latest version of PSoC Creator Images updated
*D	5965252	SNVN	01/05/2018	Updated template Code updated to work with latest version of PSoC Creator. Version of CE updated to 1.20

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Arm® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

Cypress Developer Community

Community Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2017-2018. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress does not assume any liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.