CE224714 – PSoC 6 MCU Implementing BLE Multi-connection (3 Masters 1 Slave)

Objective

This example demonstrates how to configure the PSoC® 6 MCU with Bluetooth Low Energy Connectivity (PSoC 6 BLE) device in simultaneous Multiple Master and Single Slave modes of operation.

Requirements

Tool: PSoC Creator™ 4.2; Peripheral Driver Library (PDL) 3.0.3

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

Associated Parts: All PSoC 6 MCU with BLE Connectivity parts

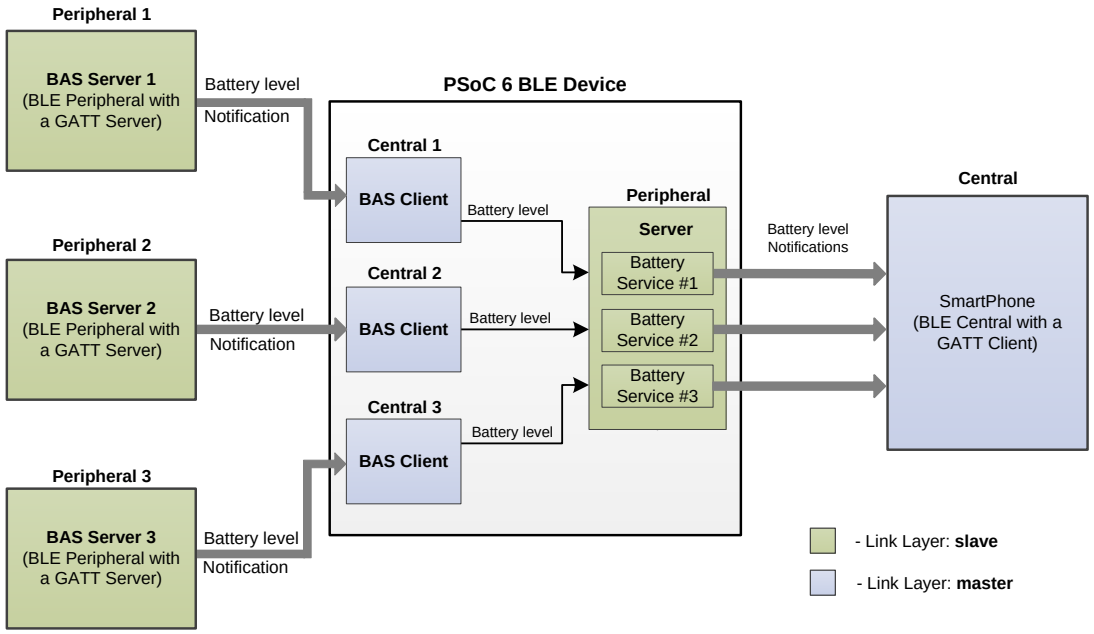
Related Hardware: CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit

Overview

The BLE Multi-Master Single Slave project is used in a pair with the CE215119 BLE Battery Level code examples for PSoC 6 MCU or PSoC 4 devices to demonstrate the operation in simultaneous Multiple Master and Single Slave modes. The Multi-Master Single Slave project uses three BLE Central connections and one Peripheral connection:

- The Central device is configured as a Generic Attribute Profile (GATT) Client with a Battery Service that can communicate with a peer device in the Generic Access Profile (GAP) Peripheral and GATT Server roles. Use the existing *CE215119 - BLE Battery Level* code examples for PSoC 6 MCU or PSoC 4 devices or an application that can simulate a GATT Server with a Battery Service as a peer device.
- The Peripheral device is configured as a GATT Server with three Battery Services. This configuration represents the battery level of the three Peripherals that the device is connected to. Figure 1 shows a block diagram of the Multi-Master Single Slave.

Figure 1. Multi-Master Single-Slave



This code example assumes that you are familiar with the PSoC 6 BLE device and the PSoC Creator integrated design environment (IDE). If you are new to PSoC 6 BLE, see the application note AN221774 – Getting Started with PSoC 6 MCU and AN210781 – Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity.

This code example uses FreeRTOS. See PSoC 6 101: Lesson 1-4 FreeRTOS training video to learn how to create a PSoC 6 FreeRTOS project with PSoC Creator. Visit the FreeRTOS website for documentation and API references of FreeRTOS.

Hardware Setup

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

Software Setup

This code example requires CySmart application. 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.

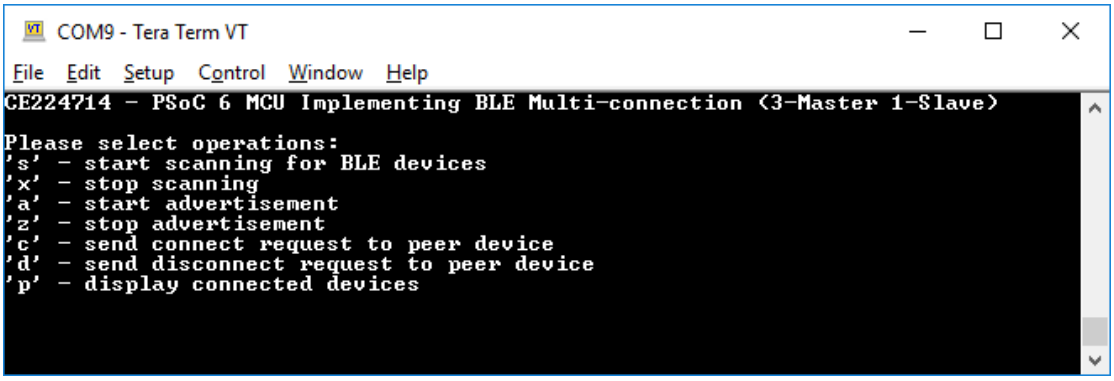
iOS Android

This code example also requires a PC terminal emulator for user interface.

Operation

1. Plug the CY8CKIT-062-BLE kit board into your computer's USB port.
2. Open terminal software such as Tera Term and select the KitProg2's COM port with a baud rate setting of 115200 bps, data bits 8, parity none, and stop bit 1.
3. 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. Do not replace the file *stdio_user.h* and *FreeRTOSConfig.h* files, if prompted by PSoC Creator.
4. Confirm that the program is working. The terminal should show a message with the available options, see Figure 2.

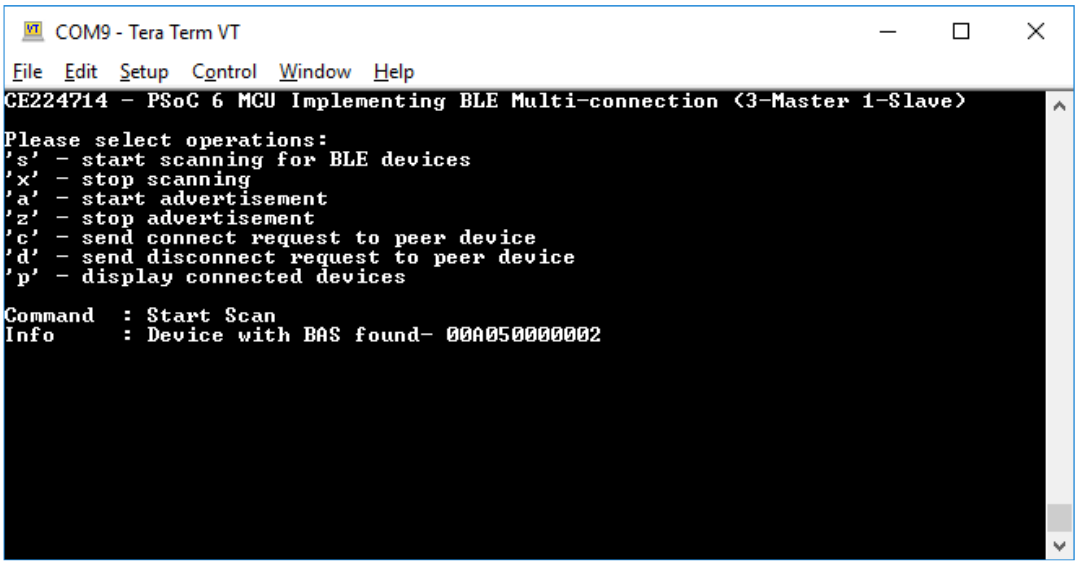
Figure 2. Terminal Application



5. Program the other PSoC 6 BLE device with *CE215119 BLE Battery Level* code example, and make sure device is advertising.

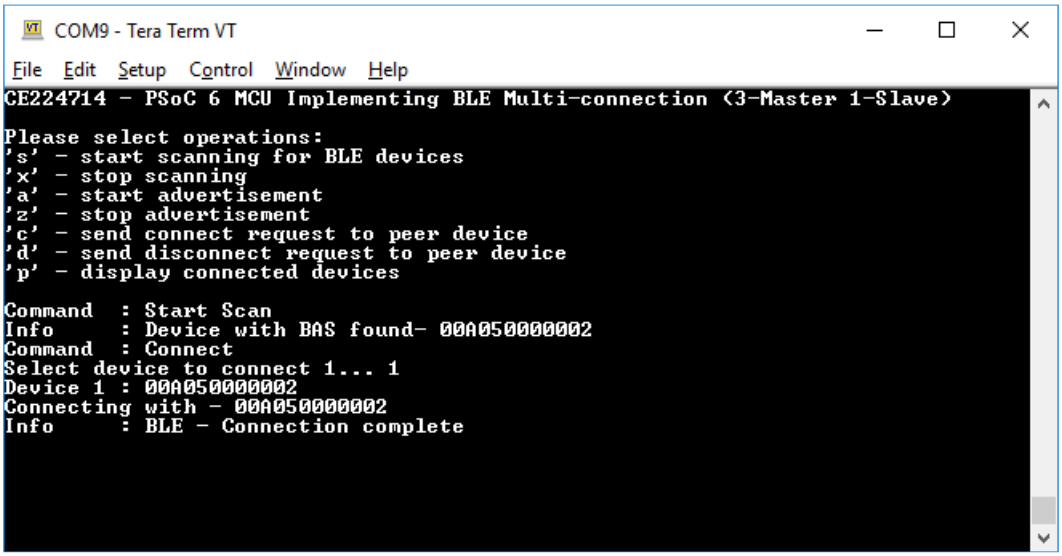
6. Press ‘s’ in the terminal application to scan for the device. All devices, with Battery Service, in the range are listed (see Figure 3).

Figure 3. Terminal Application - BLE Scan



7. Press ‘c’ to send a connection request. From the list of addresses displayed, select the number corresponding to the address of the device you want to connect to. After the successful connection, a success message is displayed as shown in Figure 4. Similarly, you can connect to a maximum of three peripheral devices.

Figure 4. Terminal application - BLE device connected



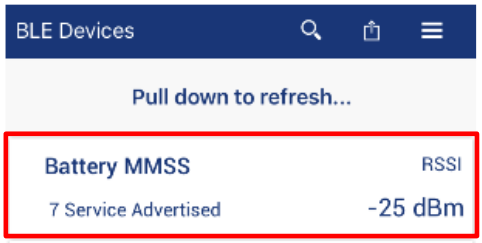
8. To emulate a Central device, you can use a personal computer running the CySmart tool or a mobile device running the CySmart mobile application. Press ‘a’ in the terminal application to star adverting.

9. Do the following to test using the CySmart mobile app:

   a. Turn ON Bluetooth on your Android or iOS device. Launch the CySmart app.

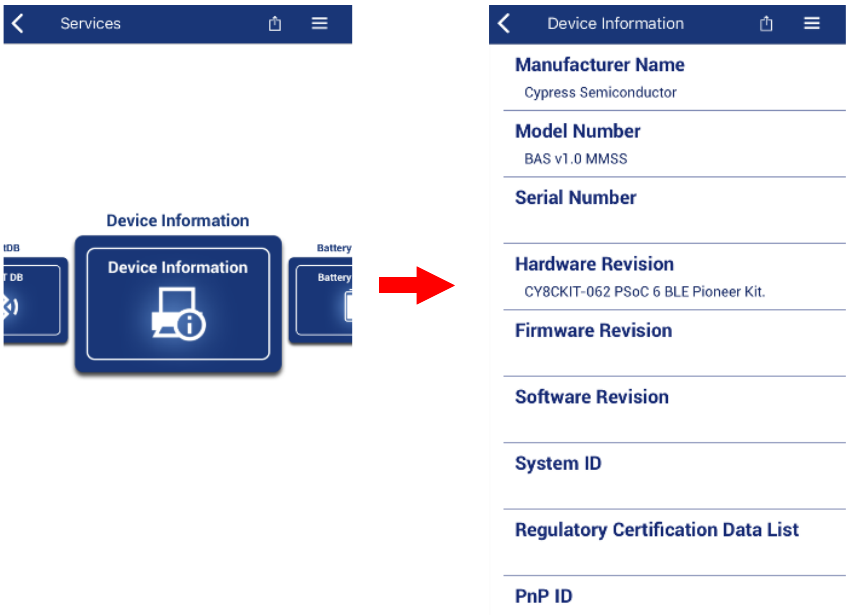
   b. Pull down the CySmart app home screen to start scanning for BLE Peripherals; your device appears in the CySmart app home screen as shown in Figure 5. Select your device to establish a BLE connection.

Figure 5. Device Selection



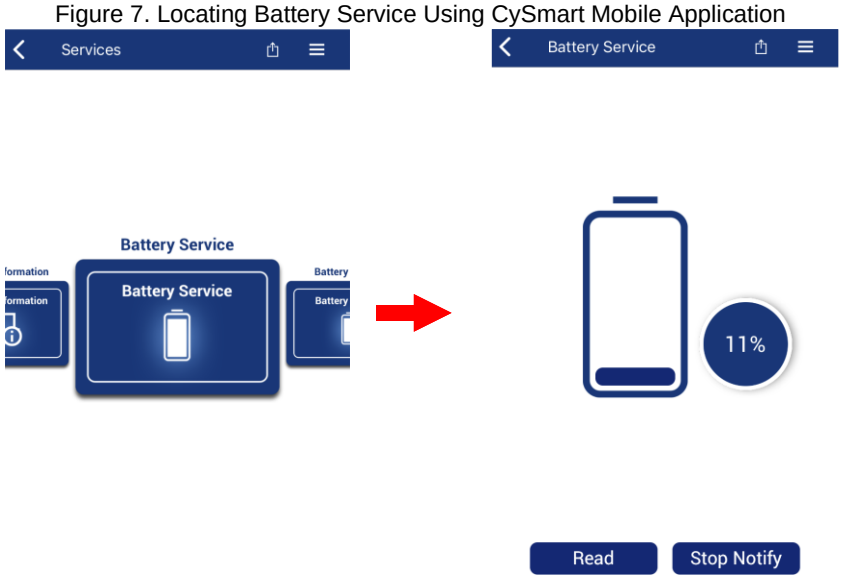
c. Select the **Device Information** profile to get the manufacturer, vendor, or both information about the device, as Figure 6 shows.

Figure 6. Locating Device Information Using CySmart Mobile Application



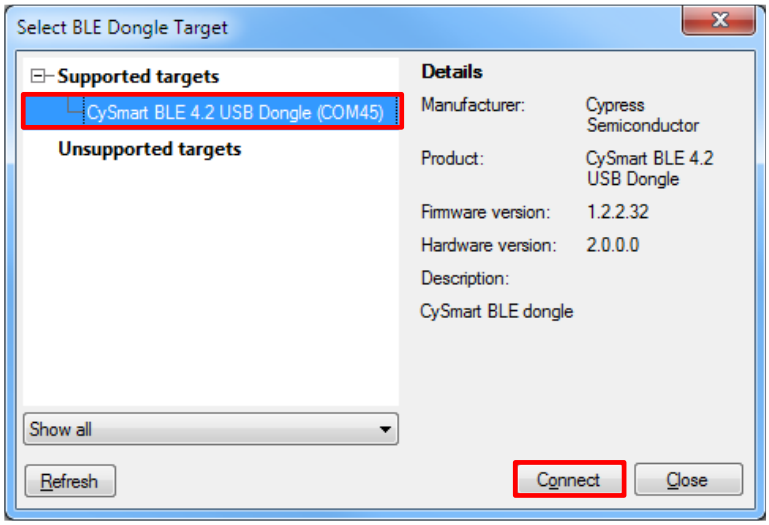
d. Select Battery Service to see the battery level of connected device.

Figure 7. Locating Battery Service Using CySmart Mobile Application



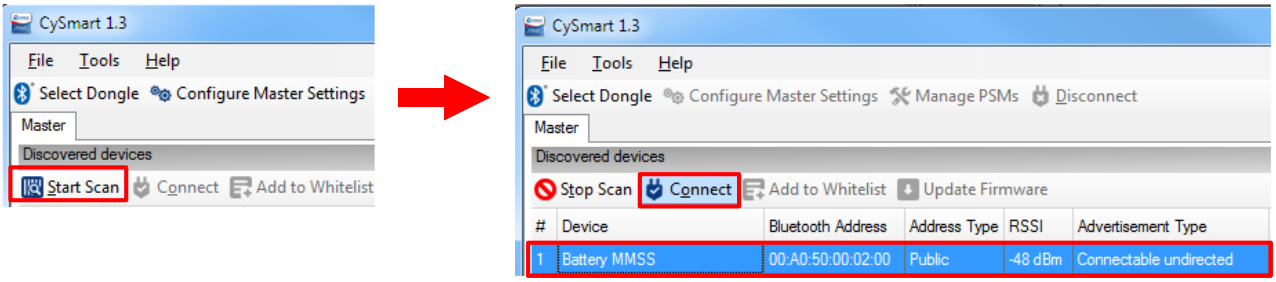
10. Do the following to test using the CySmart Host Emulation Tool:
    a. Connect the BLE Dongle to your Windows PC. Wait for the driver installation to complete, if not done already.
    b. Launch the CySmart Host Emulation Tool.
    c. CySmart automatically detects BLE Dongles connected to the PC. Click **Refresh** if the BLE Dongle does not appear in the **Select BLE Dongle Target** pop-up window. Click **Connect**, as shown in Figure 8.

Figure 8. CySmart BLE Dongle Selection



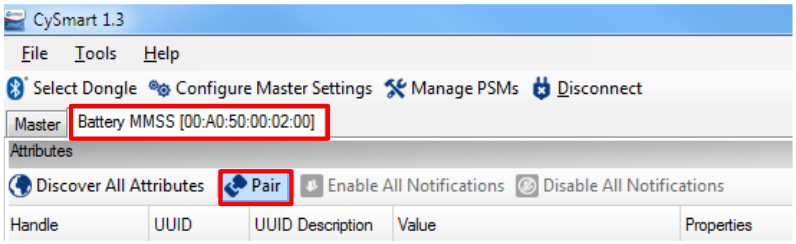
    d. Click **Start Scan** to discover available devices. Select **Battery MMSS** in the list of available devices and connect to it (see Figure 9).

Figure 9. CySmart Device Discovery and Connection



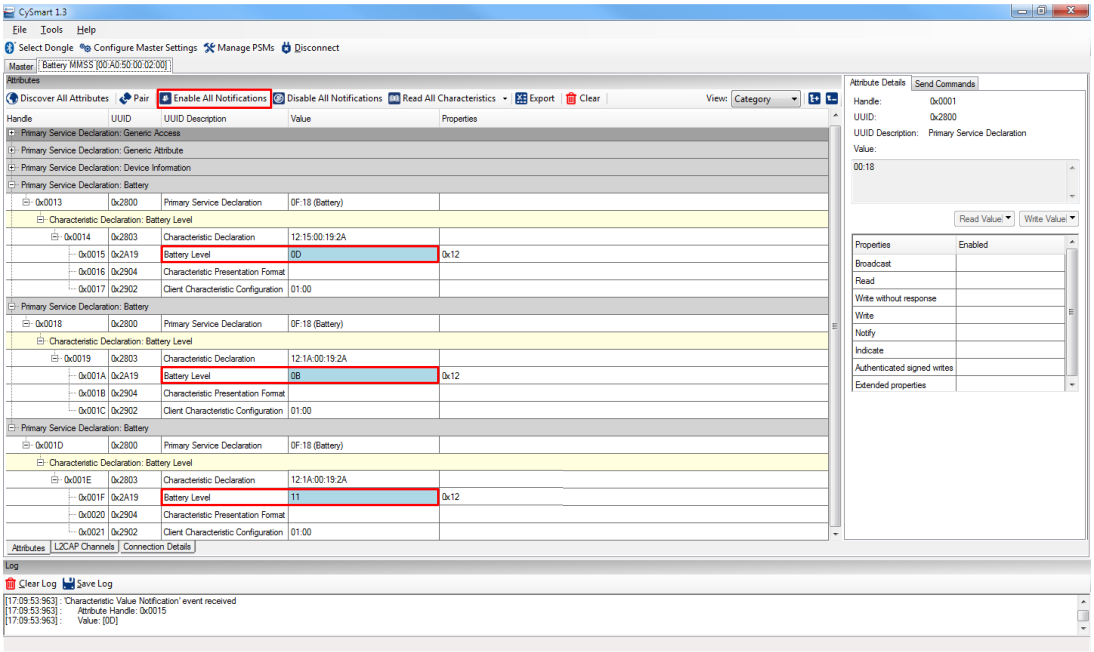
e. After connection is complete click **Pair** as shown in Figure 10.

Figure 10. CySmart Device Pairing



f. Click **Discover All Attributes**, and then **Enable All Notifications**. Observe the received characteristic values (see Figure 11).

Figure 11. CySmart Enabling all Notifications

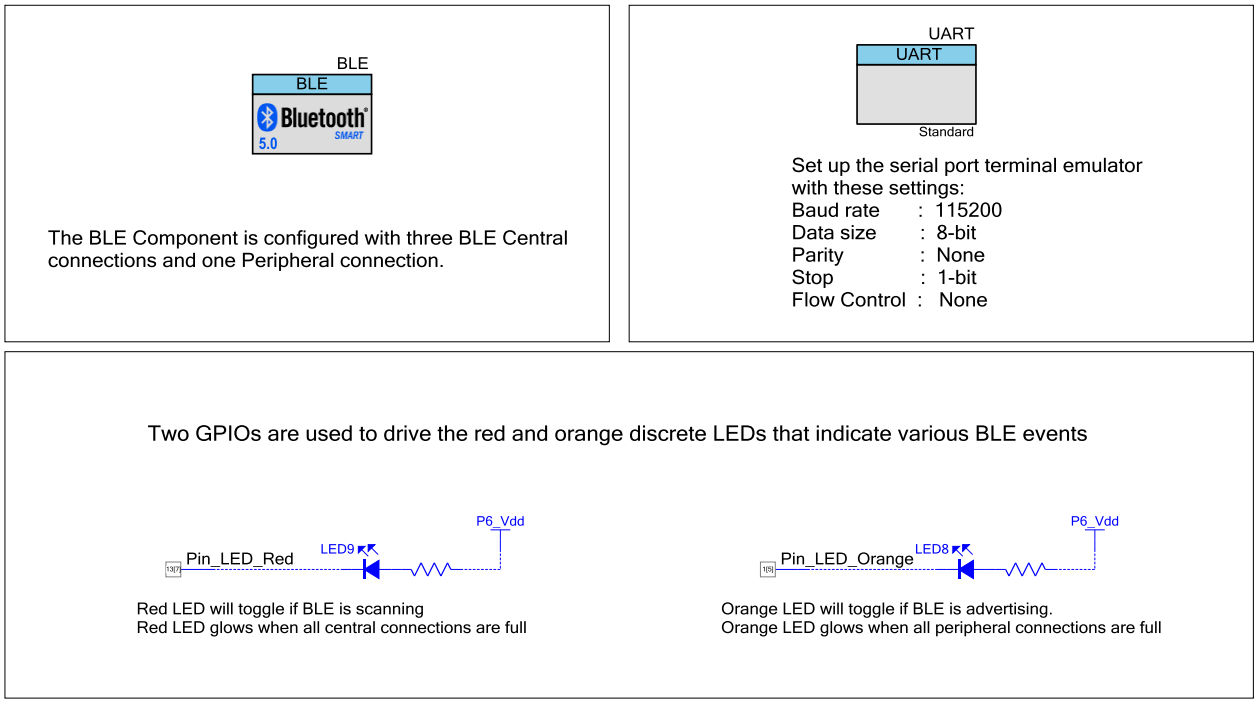


11. Set the `UART_DEBUG_ENABLE` macro in *uart_debug.h* to true and rebuild the project to see verbose debug messages.

Design and Implementation

Figure 12 shows the top design schematic.

Figure 12. BLE Multi Master Single Slave (RTOS) Code Example Schematic



This code example demonstrates how to configure the PSoC 6 BLE device in simultaneous Multiple Master and Single Slave modes of operation. In this project, there are three tasks

1. BLE task – Handles BLE host initialization and processes general BLE events. The BLE Component requires several callback functions to receive events from the BLE Stack. `StackEventHandler()`is used to receive general BLE events. `BasCallBack()` is used to receive events specific to the service's attribute operations.
2. UART task – Handles the UART interface
3. Status LED task – Controls the state of status LEDs

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	The BLE Component is configured to act as three centrals and one a peripheral.	See Parameter Setting
Digital Output Pin	Pin_LED_Red Pin_LED_Orange	These GPIOs are configured as firmware-controlled digital output pins that control LEDs.	[**General** tab] Uncheck HW connection Drive mode: Strong Drive
UART (SCB)	UART	This Component is used to print messages on a terminal program and take user input from the terminal.	Default

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

Parameter Setting

Figure 13 to Figure 21 show the BLE Component configuration.

Figure 13. BLE – General Settings

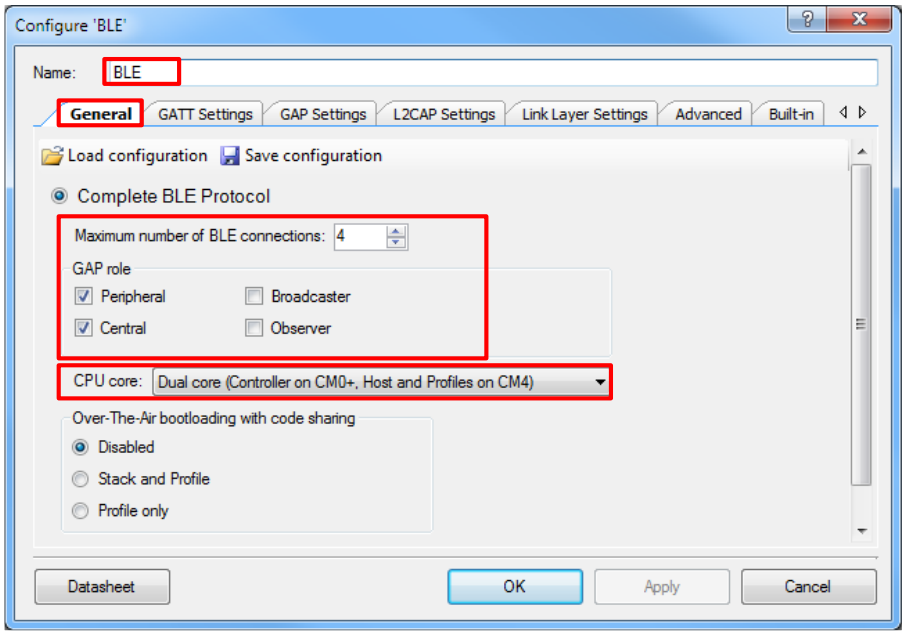


Figure 14. BLE – GATT Settings

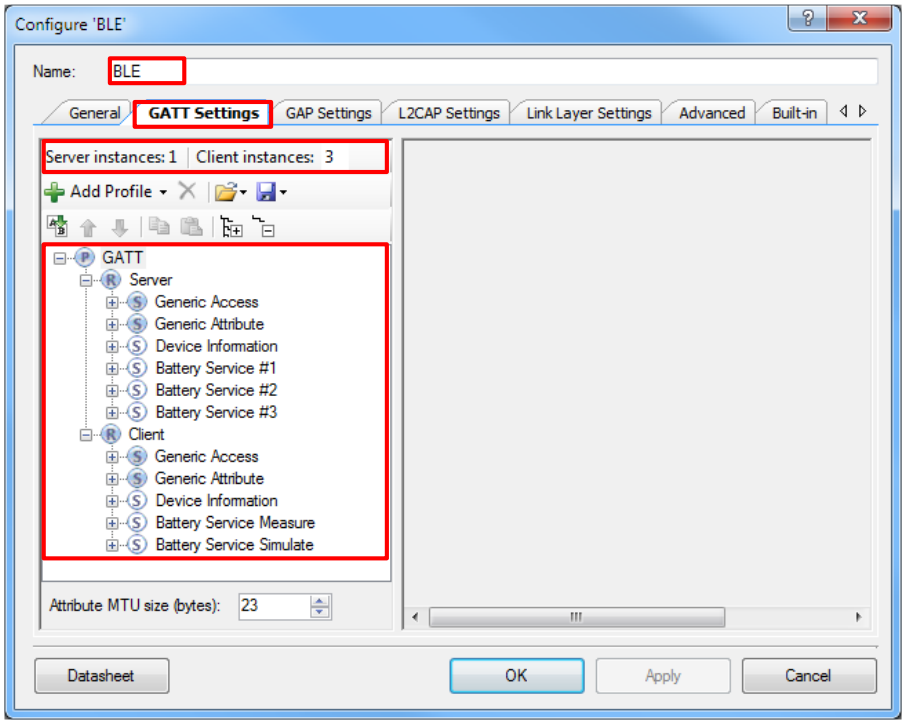


Figure 15. BLE – GAP Settings General configuration

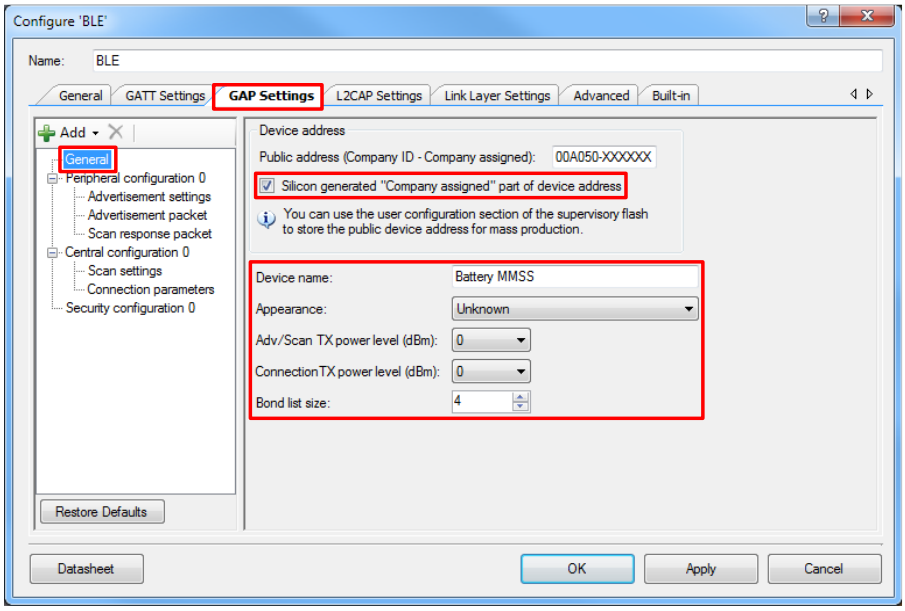


Figure 16. BLE – GAP Advertisement settings

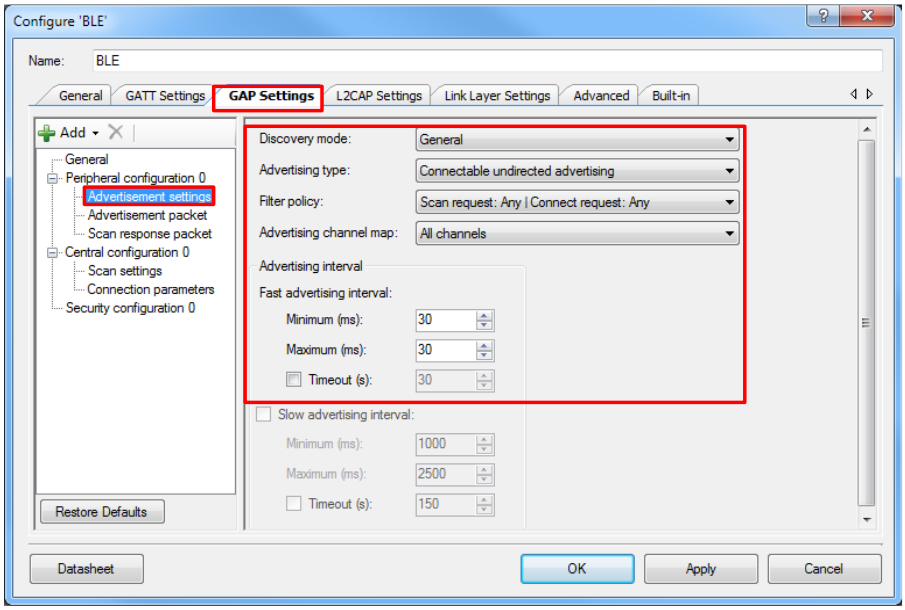


Figure 17. BLE – GAP Advertisement Packet Configuration

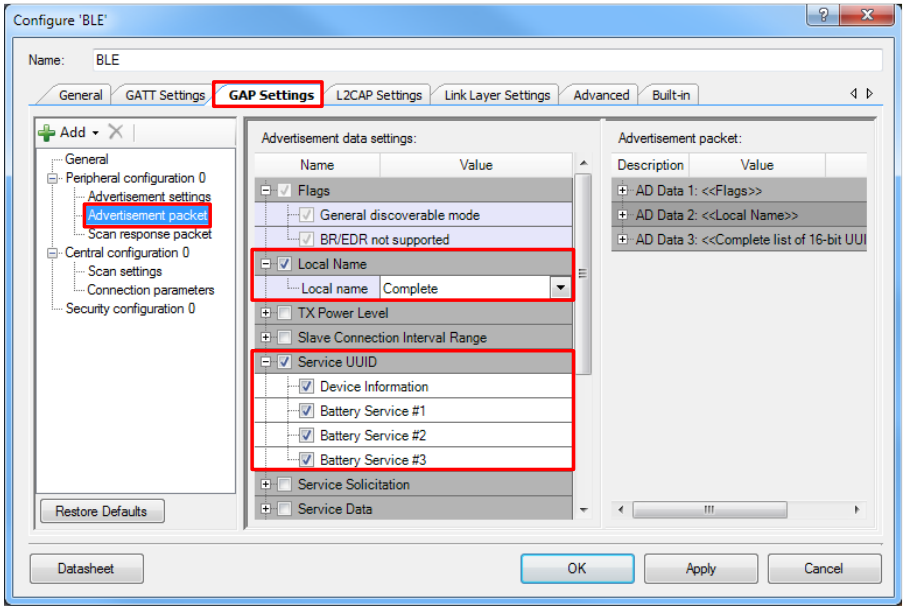


Figure 18. BLE – GAP Scan Response Packet Configuration

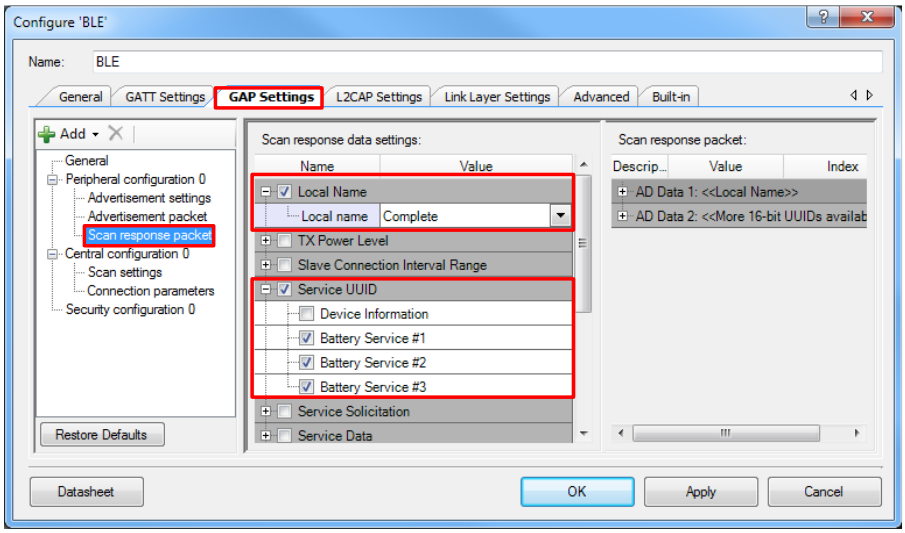


Figure 19. BLE – GAP Central Scan Settings

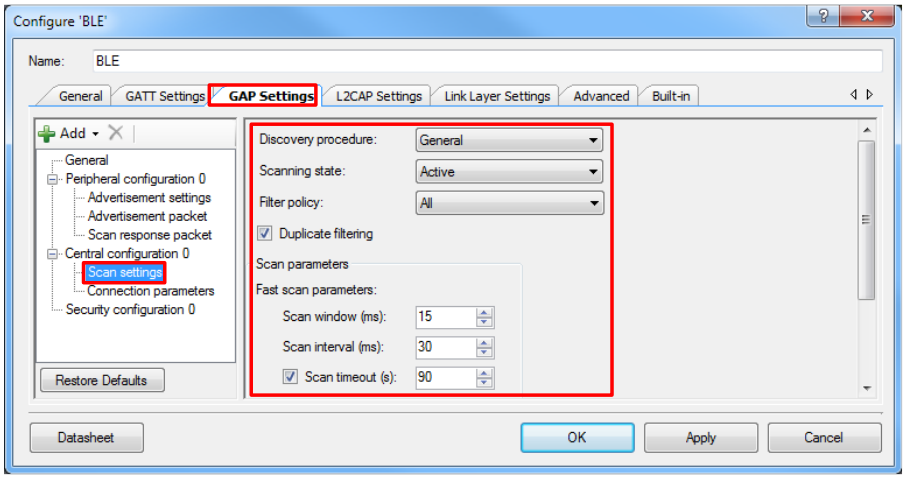


Figure 20. BLE – GAP Central Connection Parameter Configuration

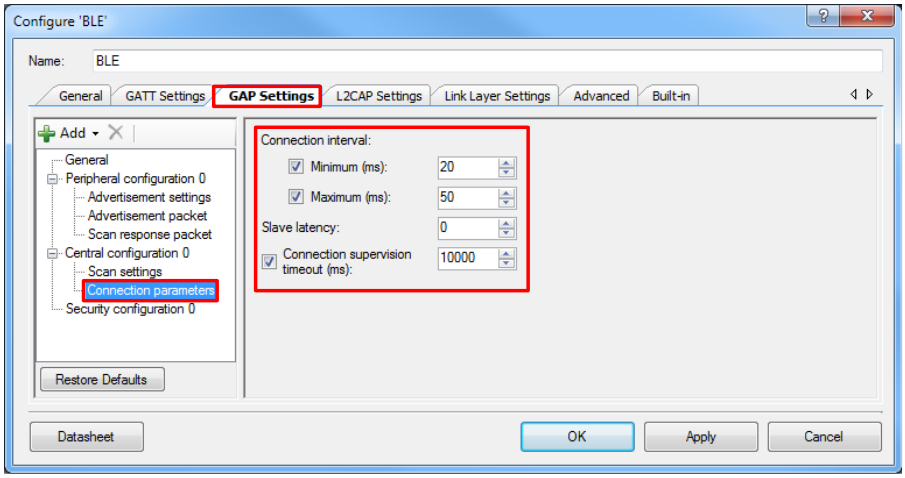


Figure 21. BLE – GAP Security Configuration

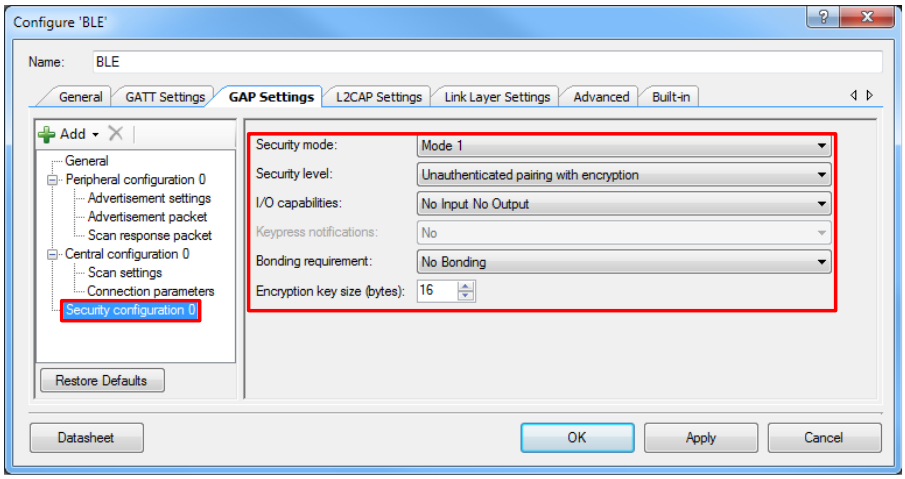
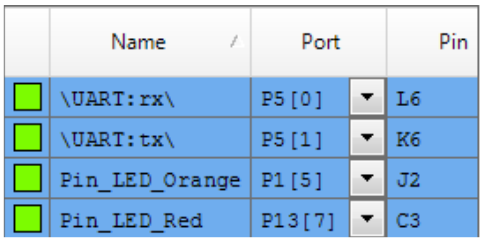


Figure 22 shows the pin assignment for the project done through the **Pins** tab in the **Design Wide Resources** window. These assignments are compatible with CY8CKIT-062-BLE.

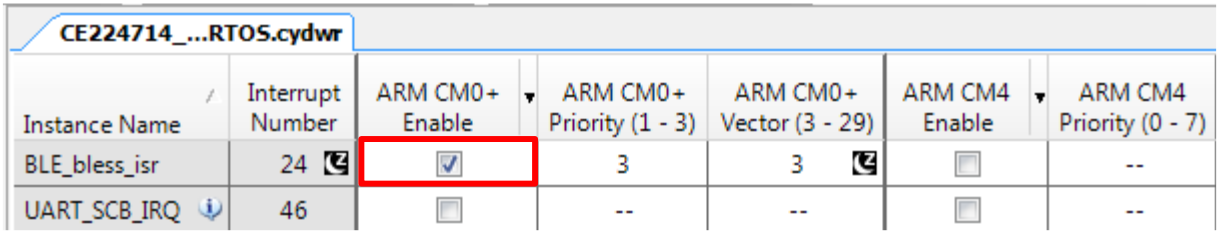
Figure 22. DWR Pin Assignment Table



	Name	Port	Pin
	\UART:rx\	P5[0]	L6
	\UART:tx\	P5[1]	K6
	Pin_LED_Orange	P1[5]	J2
	Pin_LED_Red	P13[7]	C3

Figure 23 shows the interrupt configuration for the project.

Figure 23. System Interrupt Configuration



CE224714_...RTOS.cydwr

Instance Name	Interrupt Number	ARM CM0+ Enable	ARM CM0+ Priority (1 - 3)	ARM CM0+ Vector (3 - 29)	ARM CM4 Enable	ARM CM4 Priority (0 - 7)
BLE_bless_isr	24	☑	3	3	☐	--
UART_SCB_IRQ	46	☐	--	--	☐	--

Reusing This Example

This example is designed for the CY8CKIT-062-BLE pioneer kit. To port the design to a different PSoC 6 MCU device, kit, or both, change the target device using the Device Selector and update the pin assignments in the Design Wide Resources Pins settings as needed.

In some cases, a resource used by a code example (for example, an IP 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.

Related Documents

Application Notes	
AN210781 – Getting Started with PSoC 6 MCU with BLE Connectivity	Describes PSoC 6 MCU with BLE Connectivity devices and how to build your first PSoC Creator project
AN221774 – Getting Started with PSoC 6 MCU	Describes PSoC 6 MCU devices and how to build your first PSoC Creator project
AN215656 – PSoC 6 MCU Dual-CPU System Design	Describes the dual-CPU architecture in PSoC 6 MCU, and shows how to build a simple dual-CPU 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.
Pins	Supports connection of hardware resources to physical pins
UART	Provides asynchronous serial communications

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	
Tool Documentation	
PSoC Creator	Look in the downloads tab for Quick Start and User Guides
Peripheral Driver Library (PDL)	Installed by PSoC Creator 4.2. Look in the *<PDL install folder>/doc* for the User Guide and the API Reference

Cypress Resources

Cypress provides a wealth of data at www.cypress.com to help you to select the right device, and quickly and effectively integrate the device into your design.

The following is an abbreviated list of resources related to this code example:

- **Overview:** MCU Portfolio, PSoC & MCU Roadmap
- **Product Selectors:** PSoC 1, PSoC 3, PSoC 4, PSoC 5LP, or PSoC 6. In addition, PSoC Creator includes a device selection tool.
- **Datasheets:** Describe and provide electrical specifications for MCU and PSoC device families.
- **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.
- PSoC **Technical Reference Manuals (TRM):** Provide detailed descriptions of the architecture and registers for a PSoC device family.
- **PSoC 6 MCU Training Videos**: Provide guidance on getting started.
- **CapSense Design Guides:** Learn how to design capacitive touch-sensing applications.
- **Development Kits:** Some examples include:
  - 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.
  - PSoC 6 WiFi-BT Pioneer Kit supports the PSoC 62 series MCU along with Wi-Fi and BT connectivity
  - 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.

Document History

Document Title: CE224714 – PSoC 6 MCU Implementing BLE Multi-connection (3 Masters 1 Slave)

Document Number: 002-24714

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6248713	AJYA	08/22/2018	New code example

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 | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

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