CE217647 - BLE Indoor Positioning with PSoC 6 MCU with BLE Connectivity

Objective

This Bluetooth Low Energy (BLE) example project demonstrates how to create an indoor navigation system using the BLE broadcasting mode that can be configured over a GATT connection.

Overview

This example project configures the BLE Pioneer Kit as a time-multiplexed broadcaster and a connectable Indoor Positioning Service (IPS) server. The GAP role is set to the broadcaster or peripheral and the GATT role set to the server. By default, the device broadcasts the IPS data and then switches over to Connectable Advertisement mode when a button is pressed. The IPS data broadcast interval is 100 ms, and the IPS Broadcast mode is indicated by the blue LED on the BLE Pioneer Kit. The connectable advertisement interval (to configure IPS data over GATT connection) is set to 20-30 ms for 180 seconds and the BLE device switches over to IPS Broadcast mode on an advertisement timeout or when a button is pressed. The green LED on the BLE Pioneer Kit indicates the Connectable Advertisement mode and the red LED indicates the connected state.

This example supports all the GATT sub-procedures defined in the IPS specification.

Requirements

Tool: PSoC Creator™ 4.2

Programming Language: C (Arm® GCC 5.4-2016-q2-update)

Associated Parts: All PSoC 6 BLE parts

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

Hardware Setup

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

1. Connect the BLE Pioneer Kit to the computer's USB port.
2. Connect the BLE Dongle to one of the USB ports on the computer.

LED Behavior

If the V_{DDD} voltage is set to lesser than 2.7 V in the DWR settings **System** tab, only the red LED is used. The red LED blinks to indicate that the device is advertising. The red LED is OFF when a device is connected to a peer device. When the device is in Hibernate mode, the red LED stays ON.

LED behavior for V_{DDD} voltage greater than 2.7 V is described in the Operation section.

Software Setup

BLE Host Emulation Tool

This example requires the CySmart application. 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 one of the following QR codes from your mobile phone to download the CySmart app.

iOS

Android

Terminal Tool

This example uses a terminal window. You must have terminal software, such as Tera Term or PuTTy.

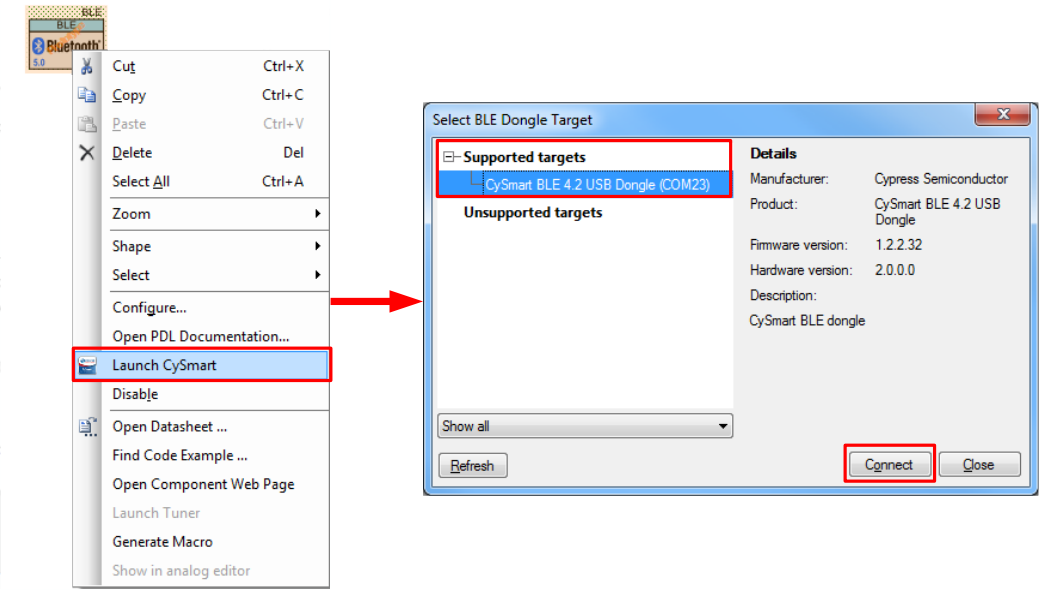
Operation

You can connect to the IPS Server device with CySmart or any BLE 4.1- or BLE 4.2-compatible device configured in the GAP Central role and capable of discovering IPS. To connect to the IPS Server, press **SW2** on CY8CKIT-062 PSoC 6 BLE Pioneer Kit to switch to Connectable Advertisement mode. The blue LED indicates that the Connectable Advertisement mode is enabled. To accept the password displayed on the HyperTerminal, press **SW2** on the CY8CKIT-062 PSoC 6 BLE Pioneer Kit or press '**y**' on the HyperTerminal. Optionally, the example project can use the legacy Security Mode 1 Level 3 (Authenticated pairing with encryption).

Operation Steps

1. Plug the CY8CKIT-062-BLE kit board into your computer's USB port.
2. Open a terminal window and perform following configuration: Baud rate – 115200, Parity – None, Stop bits – 1, Flow control – XON/XOFF. These settings must match the configuration of the PSoC Creator UART Component in the project.
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.
4. Observe the blue LED blinks while the device is advertising, and the output in the terminal window.
5. Do the following to test example, using the CySmart Host Emulation Tool application as Indoor Positioning Service Client:
    a. Connect the BLE Dongle to your Windows PC. Wait for the driver installation to complete, if necessary.
    b. Right-click the BLE Component and select Launch CySmart to launch the CySmart Host Emulation Tool. Alternatively, you can navigate to **Start** > **Programs** > **Cypress** and click **CySmart**.
    c. CySmart automatically detects the BLE dongle 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 1.

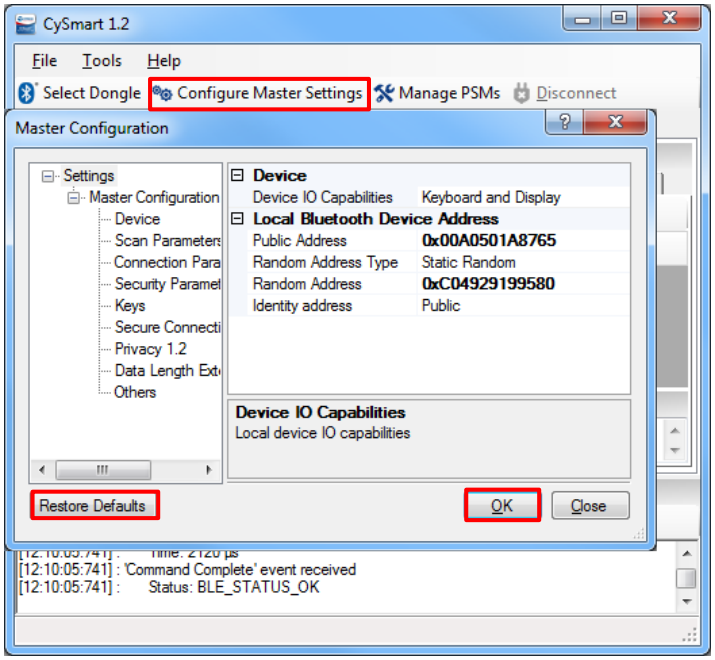
Figure 1. CySmart BLE Dongle Selection



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

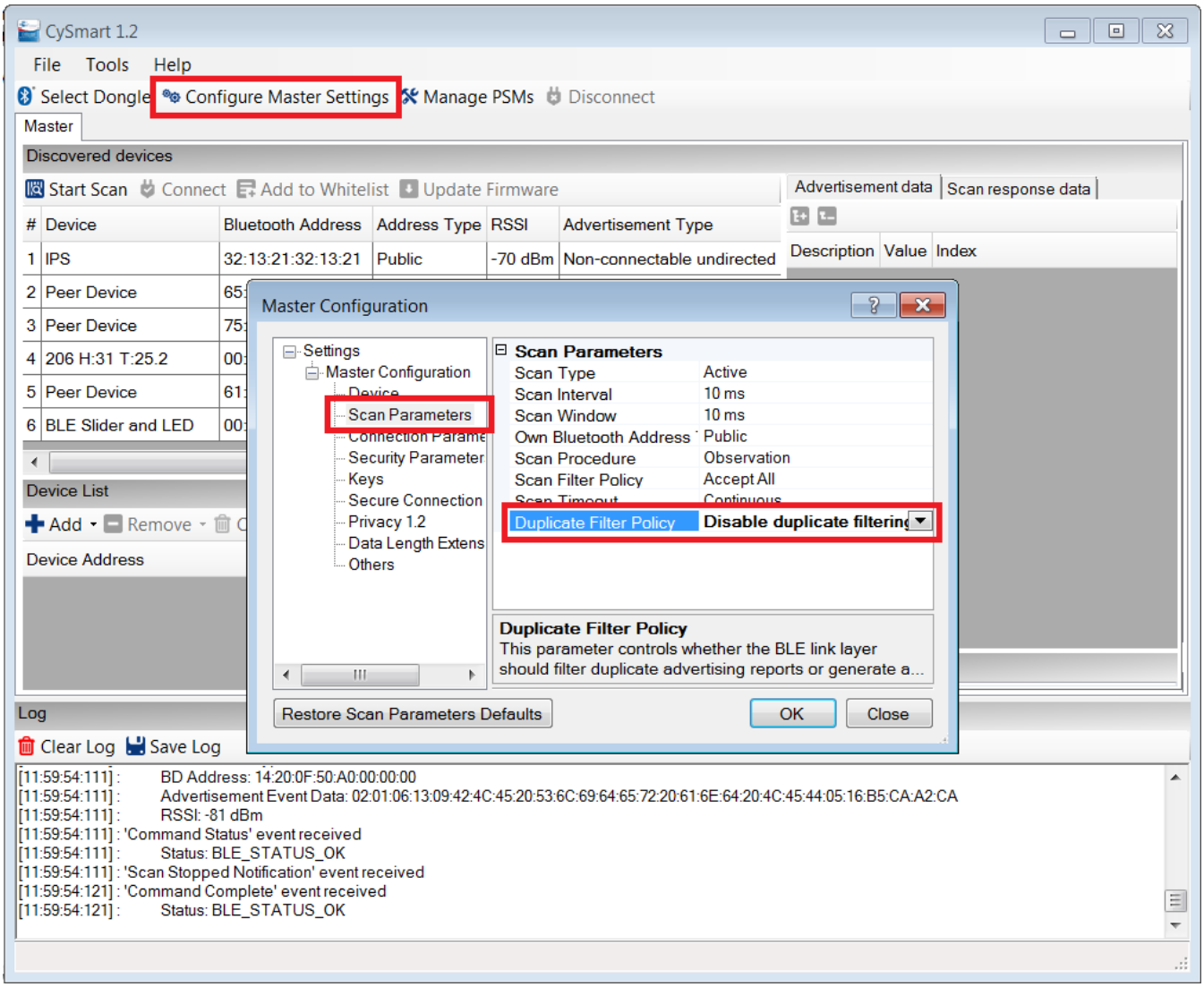
d. Select **Configure Master Settings** and then click **Restore Defaults**, as Figure 2 shows. Then click **OK**.

Figure 2. CySmart Master Settings Configuration



e. Set the **Duplicate Filter Policy = Disable duplicate filtering** in the **Master Configuration > Scan parameters** window. See Figure 3.

Figure 3. Master Configuration → Scan Parameters



f. Press the reset switch on the Pioneer Kit to start BLE advertisement if no device is connected or device is in Hibernate mode (red LED is on). Otherwise, skip this step.

g. Observe the simulated Latitude and Longitude values in the HyperTerminal program.

h. On the CySmart Host Emulation Tool, click **Start Scan**. Your device name (configured as **IPS**) should appear in the Discovered devices list, as Figure 4 shows.

Figure 4. CySmart Device Discovery and Connection

i. Select the device and observe the advertisement data in the **Raw Data** and **Log** windows (Figure 5). Advertisement data contains values of all Indoor Positioning Service characteristics, defined in Indoor Positioning Service Specification. The values of Latitude and Longitude are saved in a specific format described in Indoor Positioning Service Specification. The accordance between these formats is listed in Table 1.

Figure 5. CySmart Window

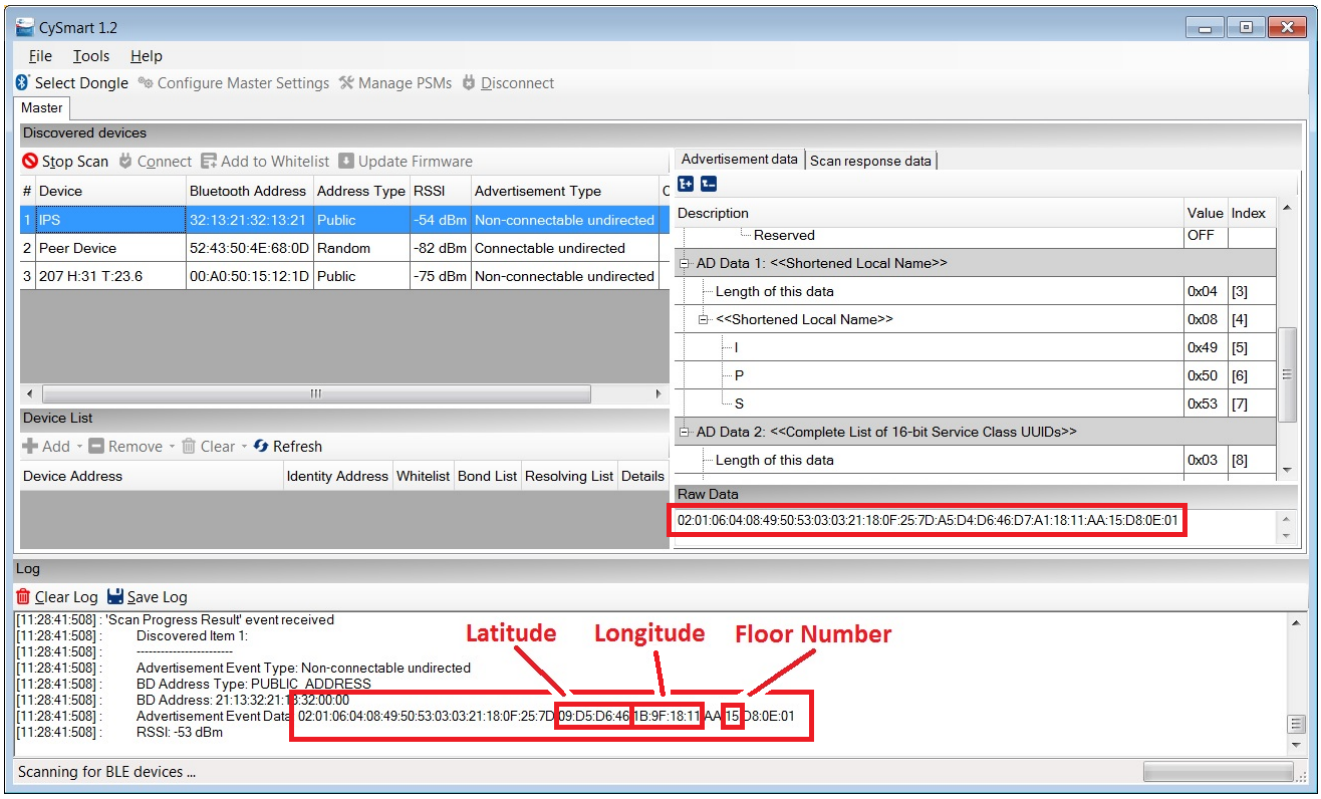


Table 1. Accordance Between Different Formats of Latitude and Longitude

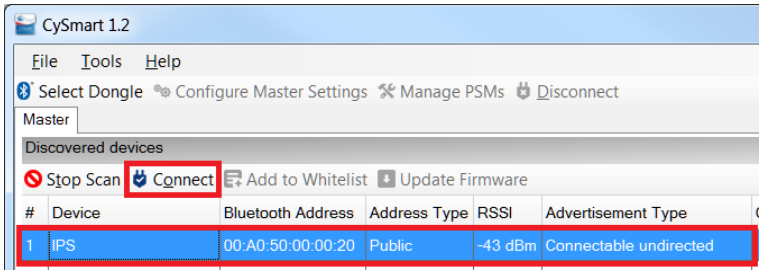
Latitude		Longitude	
49.808800	0x46D6D4A5	24.041500	0x11189F1B
49.808804	0x46D6D509	24.041508	0x11189F7F
49.808808	0x46D6D56D	24.041517	0x11189FE3
49.808813	0x46D6D5D1	24.041525	0x1118A047
49.808817	0x46D6D635	24.041533	0x1118A0AB
49.808821	0x46D6D699	24.041542	0x1118A10F
49.808825	0x46D6D6FD	24.041550	0x1118A173
49.808829	0x46D6D761	24.041559	0x1118A1D7
49.808834	0x46D6D7C5	24.041567	0x1118A23B
49.808838	0x46D6D829	24.041575	0x1118A29F
49.808842	0x46D6D88D	24.041584	0x1118A303
49.808846	0x46D6D8F1	24.041592	0x1118A367
49.808850	0x46D6D955	24.041601	0x1118A3CB

Latitude		Longitude	
49.808854	0x46D6D9B9	24.041609	0x1118A42F
49.808859	0x46D6DA1D	24.041617	0x1118A493
49.808863	0x46D6DA81	24.041626	0x1118A4F7
49.808867	0x46D6DAE5	24.041634	0x1118A55B
49.808871	0x46D6DB49	24.041642	0x1118A5BF
49.808875	0x46D6DBAD	24.041651	0x1118A623
49.808880	0x46D6DC11	24.041659	0x1118A687
49.808884	0x46D6DC75	24.041668	0x1118A6EB

j. Press **SW2** on the BLE Pioneer kit to set Connectable Advertisement mode for the Indoor Positioning Service. The green LED on the BLE Pioneer kit indicates this mode.

k. On the CySmart Host Emulation Tool, click **Stop Scan** and click **Start Scan**. Your device name (**IPS**) should appear in the Discovered devices list, as Figure 6 shows. Select the device and click **Connect** to establish a BLE connection between the CySmart Host Emulation Tool and your device.

Figure 6. CySmart Device Discovery and Connection

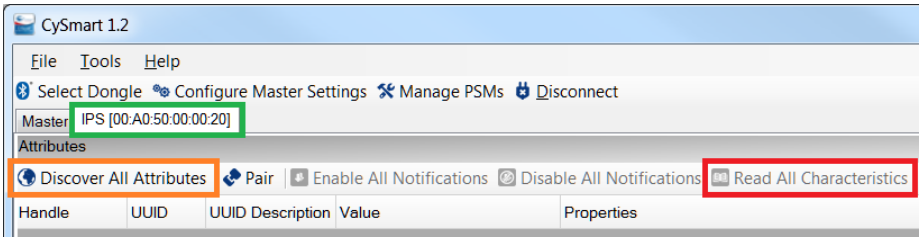


l. Click **Pair**. Click **Yes** to a pairing request received from the peer device.

m. Enter the passkey, displayed in the CySmart window, in the terminal window to confirm the Numeric comparison pairing procedure.

n. Click **Discover All Attributes**, then click **Read All Characteristics** in the CySmart application. Observe the received characteristic values.

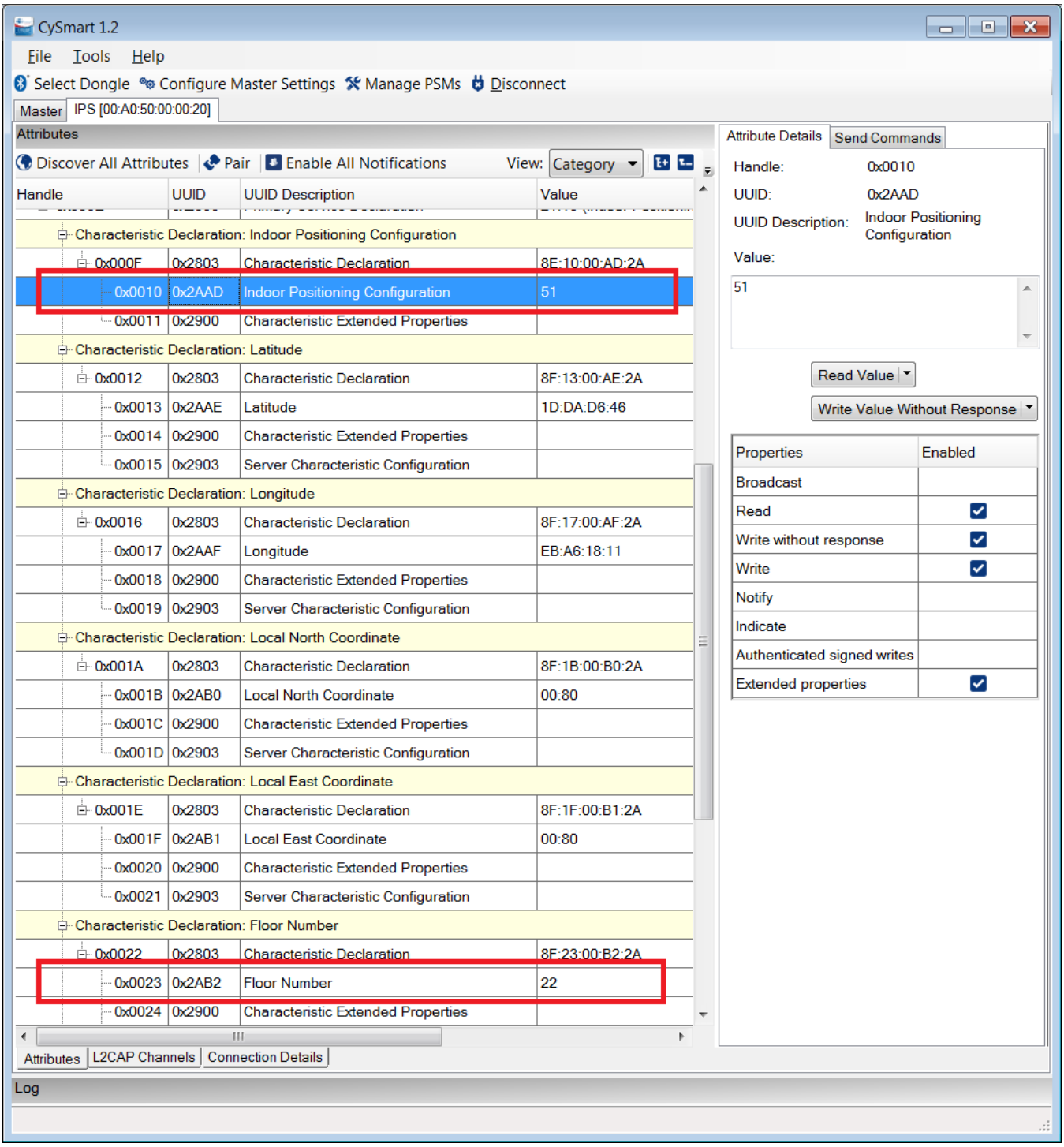
Figure 7. CySmart Attribute Discovery and Characteristics Read



o. Change the Floor Number characteristic value to 22 (for example) and click **Write Value**, then click **Read Value**. Observe the changes in CySmart and HyperTerminal. See Figure 8.

q. Change the Indoor Positioning Configuration characteristic value to 51 and click **Write Value**, then click **Read Value**. Observe the result in CySmart and HyperTerminal (Figure 8). Value 51 sets only the Latitude, Longitude, and Floor Number in the advertisement packet. For details, see Indoor Positioning Service Specification.

Figure 8. Value Writing



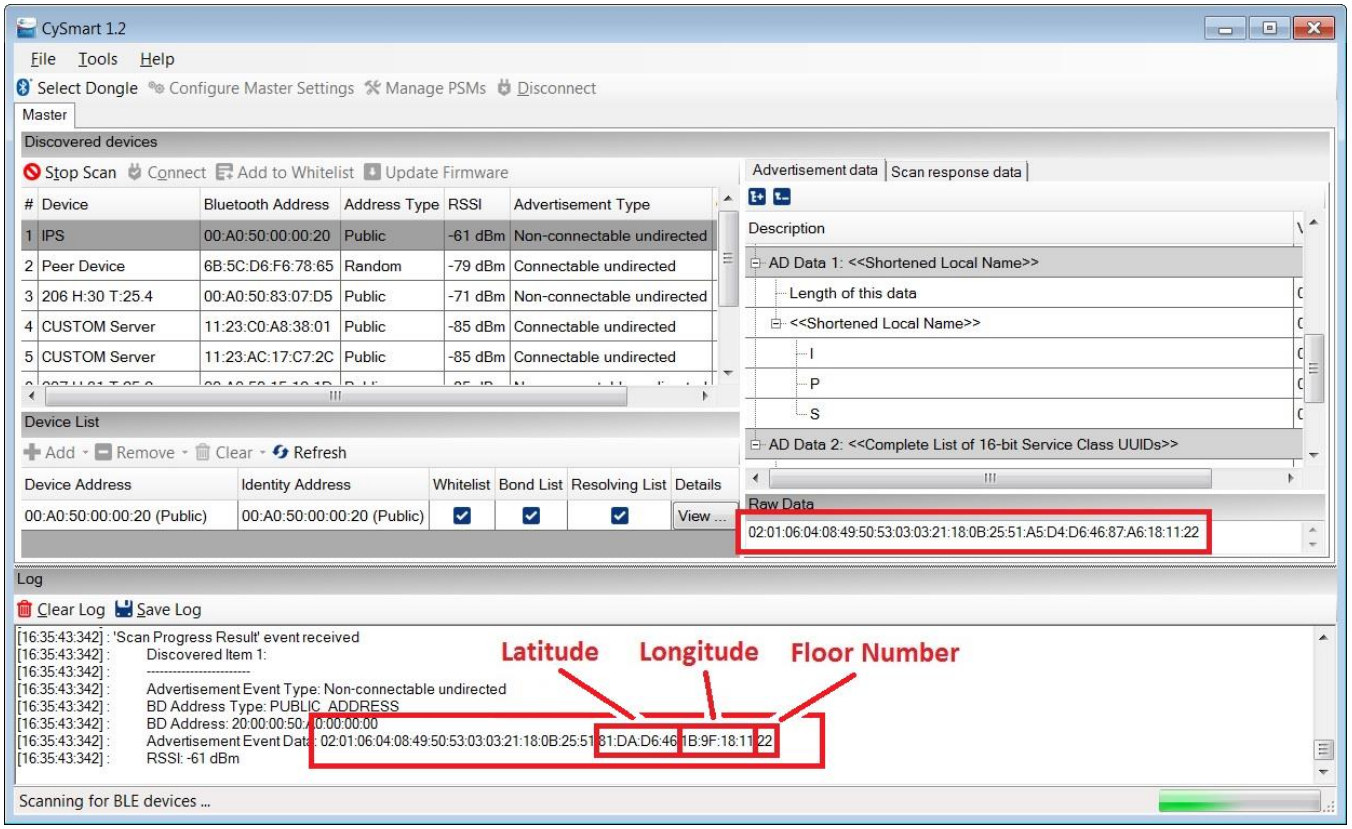
q. Click **Disconnect**, then click **Start Scan** to discover available devices.

r. Select **IPS** from the list of available devices.

s. Observe the charged advertisement packet and data in the **Raw Data** and **Log** windows. The packet contains only the values of Latitude, Longitude, and Floor Number, as set earlier.

If you have problems with using the CySmart Central Emulation Tool, see the CySmart User Guide.

Figure 9. Advertisement Packet



6. The CySmart mobile app (Android/iOS) does not have Indoor Positioning Service implementation, but still can be used in the GATT Data Base mode for testing this example. You can repeat the test flow for CySmart mobile app mentioned in step 5. For more details. see the Android and iOS CySmart User Guide.

7. Use the UART debug port to view verbose messages:

   a. The code example ships with the UART debug port enabled. To disable it, set the macro `DEBUG_UART_ ENABLED` in *common.h* to `DISABLED` and rebuild the code.

   b. The output of the debug serial port looks like the sample below:

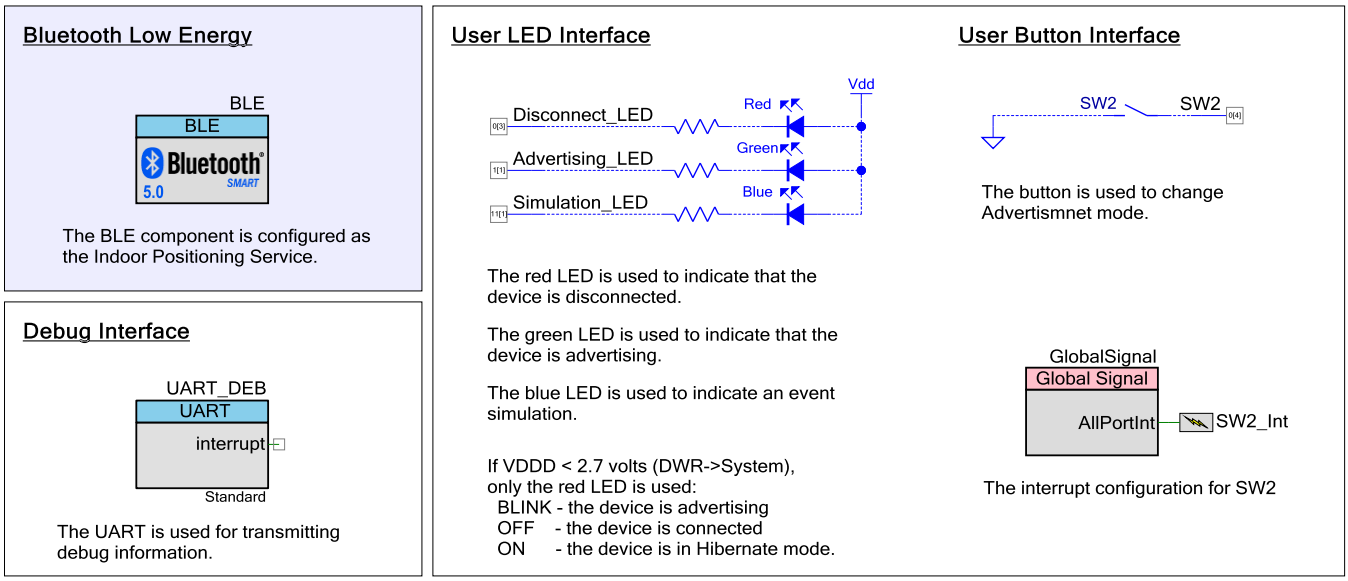
```
BLE Indoor Positioning Service Code Example
CY_BLE_EVT_STACK_ON, StartAdvertisement
CY_BLE_EVT_SET_DEVICE_ADDR_COMPLETE
CY_BLE_EVT_LE_SET_EVENT_MASK_COMPLETE
CY_BLE_EVT_GET_DEVICE_ADDR_COMPLETE: 00a050000020
CY_BLE_EVT_SET_TX_PWR_COMPLETE
CY_BLE_EVT_SET_TX_PWR_COMPLETE
CY_BLE_EVT_GAPP_ADVERTISEMENT_START_STOP, state: 2
Latitude - 49.808804. Longitude - 24.041500
CY_BLE_EVT_GAP_KEYS_GEN_COMPLETE
Latitude - 49.808808. Longitude - 24.041500
Latitude - 49.808813. Longitude - 24.041500
Latitude - 49.808817. Longitude - 24.041500
Latitude - 49.808821. Longitude - 24.041500
```

```
CY_BLE_EVT_GAPP_ADVERTISEMENT_START_STOP, state: 0
Connectable mode.
CY_BLE_EVT_GAPP_ADVERTISEMENT_START_STOP, state: 2
Latitude - 49.808825. Longitude - 24.041500
Latitude - 49.808829. Longitude - 24.041500
Latitude - 49.808834. Longitude - 24.041500
Latitude - 49.808838. Longitude - 24.041500
CY_BLE_EVT_GATT_CONNECT_IND: 0, 10
CY_BLE_EVT_GAP_DEVICE_CONNECTED: connIntv = 7 ms
CY_BLE_EVT_GATTS_XCNHG_MTU_REQ
CY_BLE_EVT_GATTS_READ_CHAR_VAL_ACCESS_REQ: handle: 3
CY_BLE_EVT_GAP_AUTH_REQ: bdHandle=10, security=3, bonding=1, ekeySize=10, err=0
CY_BLE_EVT_GAP_SMP_NEGOTIATED_AUTH_INFO: bdHandle=10, security=2, bonding=1, ekeySize=10, err=0
CY_BLE_EVT_GAP_PASSKEY_ENTRY_REQUEST
Please enter the passkey displayed on the peer device:
Enter 6 digit passkey:
3
3
0
6
0
2
CY_BLE_EVT_STACK_BUSY_STATUS: 1
CY_BLE_EVT_GAP_ENCRYPT_CHANGE: 0
CY_BLE_EVT_STACK_BUSY_STATUS: 0
CY_BLE_EVT_GAP_KEYINFO_EXCHNGE_CMPLT
CY_BLE_EVT_GAP_AUTH_COMPLETE: security:2, bonding:1, ekeySize:10, authErr 0
CY_BLE_EVT_PENDING_FLASH_WRITE
Store bonding data, status: 140001, pending: 1
Store bonding data, status: 140001, pending: 1
Store bonding data, status: 140001, pending: 1
Store bonding data, status: 0, pending: 0
CY_BLE_EVT_GATTS_READ_CHAR_VAL_ACCESS_REQ: handle: 3
CY_BLE_EVT_GATTS_READ_CHAR_VAL_ACCESS_REQ: handle: 5
CY_BLE_EVT_GATTS_READ_CHAR_VAL_ACCESS_REQ: handle: 7
CY_BLE_EVT_GATTS_READ_CHAR_VAL_ACCESS_REQ: handle: e
CY_BLE_EVT_GATTS_READ_CHAR_VAL_ACCESS_REQ: handle: 11
CY_BLE_EVT_GATTS_READ_CHAR_VAL_ACCESS_REQ: handle: 15
CY_BLE_EVT_GATTS_READ_CHAR_VAL_ACCESS_REQ: handle: 19
CY_BLE_EVT_GATTS_READ_CHAR_VAL_ACCESS_REQ: handle: 1d
CY_BLE_EVT_GATTS_READ_CHAR_VAL_ACCESS_REQ: handle: 21
CY_BLE_EVT_GATTS_READ_CHAR_VAL_ACCESS_REQ: handle: 25
CY_BLE_EVT_GATTS_READ_CHAR_VAL_ACCESS_REQ: handle: 29
CY_BLE_EVT_GATTS_READ_CHAR_VAL_ACCESS_REQ: handle: 2d
Write characteristic Floor Number. Value - 22 (34).   Len - 1.
CY_BLE_EVT_GATTS_READ_CHAR_VAL_ACCESS_REQ: handle: 21
CY_BLE_EVT_GATTS_READ_CHAR_VAL_ACCESS_REQ: handle: 21
Write characteristic Ind Pos Config. Value - 51 (81).   Len - 1.
CY_BLE_EVT_GATTS_READ_CHAR_VAL_ACCESS_REQ: handle: e
CY_BLE_EVT_GATT_DISCONNECT_IND: 0, 10
CY_BLE_EVT_GAP_DEVICE_DISCONNECTED: bdHandle=10, reason=13, status=0
CY_BLE_EVT_GAPP_ADVERTISEMENT_START_STOP, state: 2
Latitude - 49.808842. Longitude - 24.041500
Latitude - 49.808846. Longitude - 24.041500
Latitude - 49.808850. Longitude - 24.041500
Latitude - 49.808854. Longitude - 24.041500
Latitude - 49.808859. Longitude - 24.041500
```

Design and Implementation

Figure 10 shows the top design schematic.

Figure 10. BLE Indoor Positioning Code Example Schematic



This project demonstrates the functionality of the BLE Component configured as the IPS Server. It is designed to work with CySmart.

After startup, the device initializes the BLE Component. To operate, the Component requires several callback functions to receive events from the BLE Stack. `AppCallBack()` is used to receive general BLE events. Another callback (`IpsCallBack()`) is used to receive events specific to the service's attribute operations.

The `CYBLE_EVT_STACK_ON` event indicates successful initialization of the BLE Stack. After this event is received, the Component starts fast advertising with the packet structure as configured in the BLE Component Customizer.

UART is used to print the debug information and scan the commands from the terminal.

Pin Assignments

Table 2 lists the pin assignments and connections required on the development board for supported kits.

Table 2. Pin Assignment

Pin Name	Development Kit	Comment
	CY8CKIT-062	
\UART_DEB:rx\	P5[0]	
\UART_DEB:tx\	P5[1]	
\UART_DEB:rts\	P5[2]	
\UART_DEB:cts\	P5[3]	
Disconnect_LED	P0[3]	The red color of the RGB LED
Advertising_LED	P1[1]	The green color of the RGB LED
Simulation_LED	P11[1]	The blue color of the RGB LED
SW2	P0[4]	

Components and Settings

Table 3 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 3. PSoC Creator Components

Component	Instance Name	Purpose	Non-default Settings
Bluetooth Low Energy (BLE)	BLE	The BLE component is configured as the Indoor Positioning Service.	See the Parameter Settings section
Digital Input Pin	SW2	This pin is used to generate interrupts when the user button (**SW2**) is pressed.	**[General tab]** Uncheck HW connection Drive mode: Resistive Pull Up
Digital Output pin	Disconnect_LED Advertising_LED Simulation_LED	These GPIOs are configured as firmware-controlled digital output pins that control LEDs.	**[General tab]** Uncheck HW connection Drive mode: Strong Drive
SysInt	SW2_Int	This Component is configured to extract interrupts from GlobalSignal.	Default
GSRef	GlobalSignal	This Component is used to detect if any of the interrupt enabled pins triggered an interrupt. It is a separate resource from the dedicated port interrupts, and it has the ability to wake up the chip from deep-sleep mode	**[General tab]** Global signal name: HWCombined Port Interrupt (AllPortInt)
UART (SCB)	UART_DEBUG	This Component is used to print messages on a terminal program.	Default

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

Parameter Settings

The BLE Component is configured as the IPS Server in the GAP Peripheral role with the settings shown in Figure 11 to Figure 18. The Custom Profile is used as there is no defined Indoor Positioning Profile specification defined.

Figure 11. General Settings

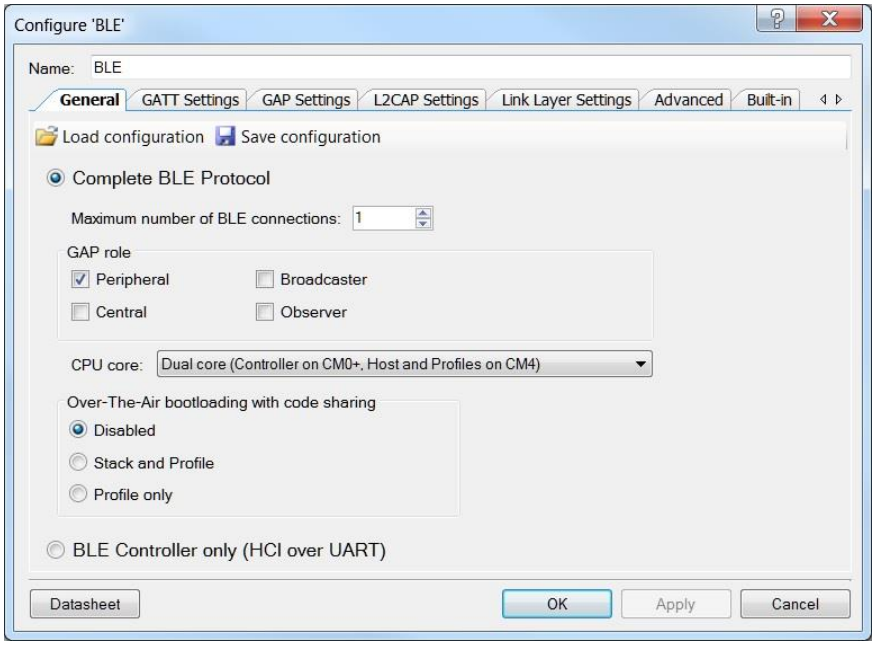


Figure 12. GATT Settings

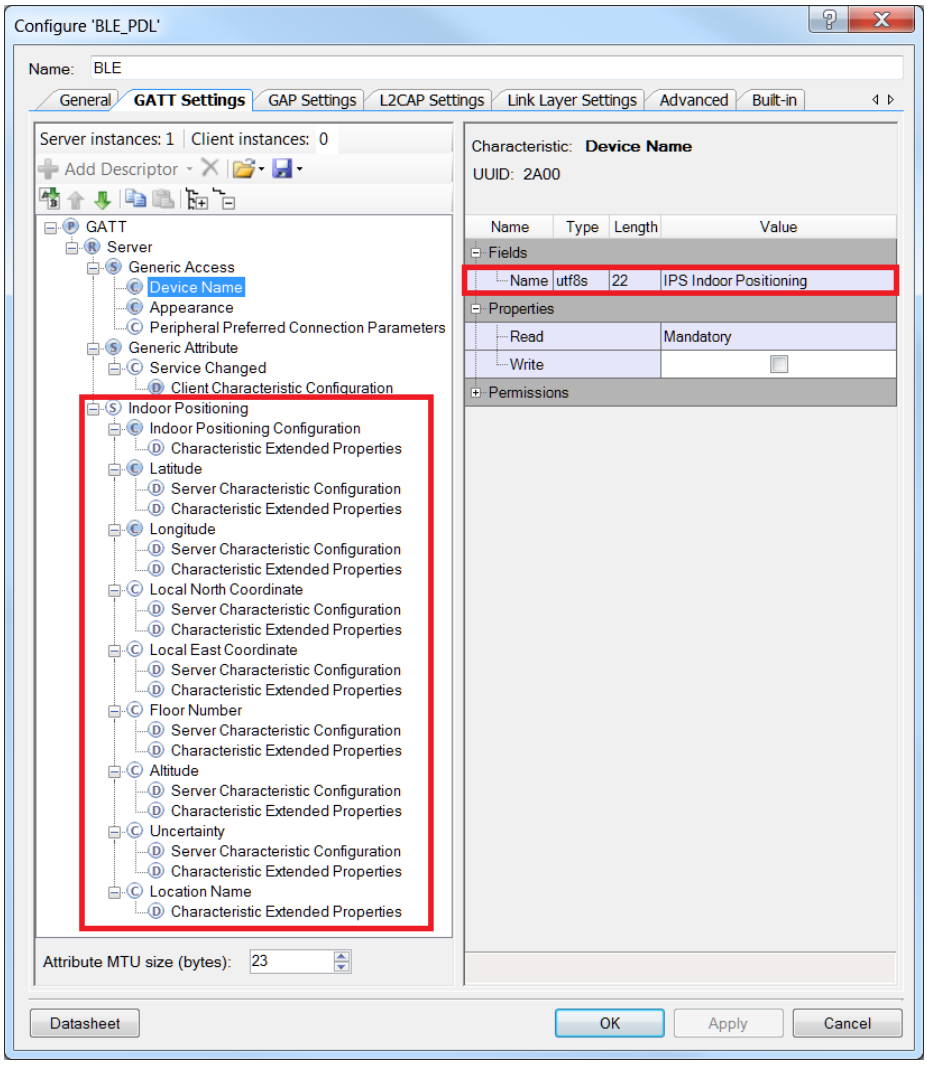


Figure 13. GAP Settings

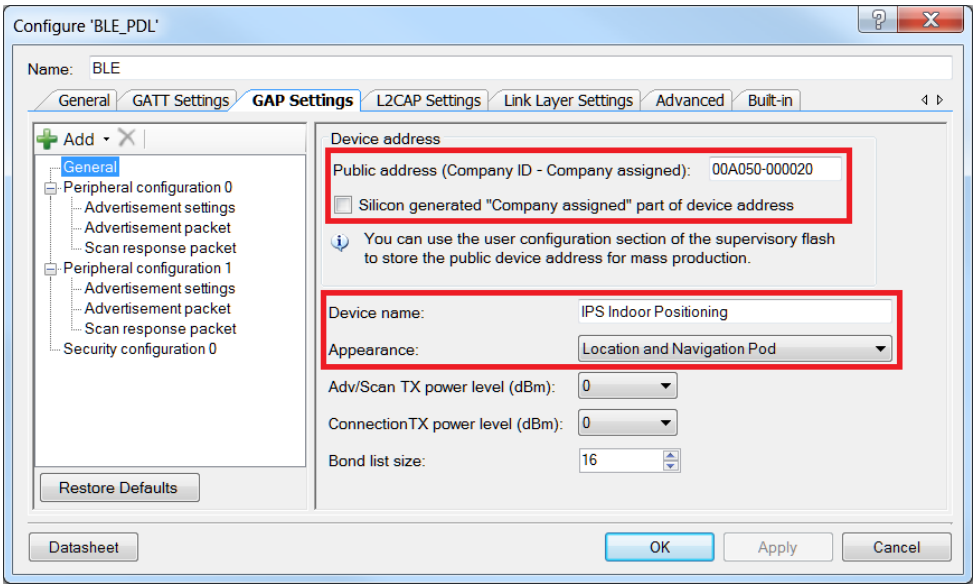


Figure 14. GAP Settings: Peripheral configuration 0 → Advertisement Settings

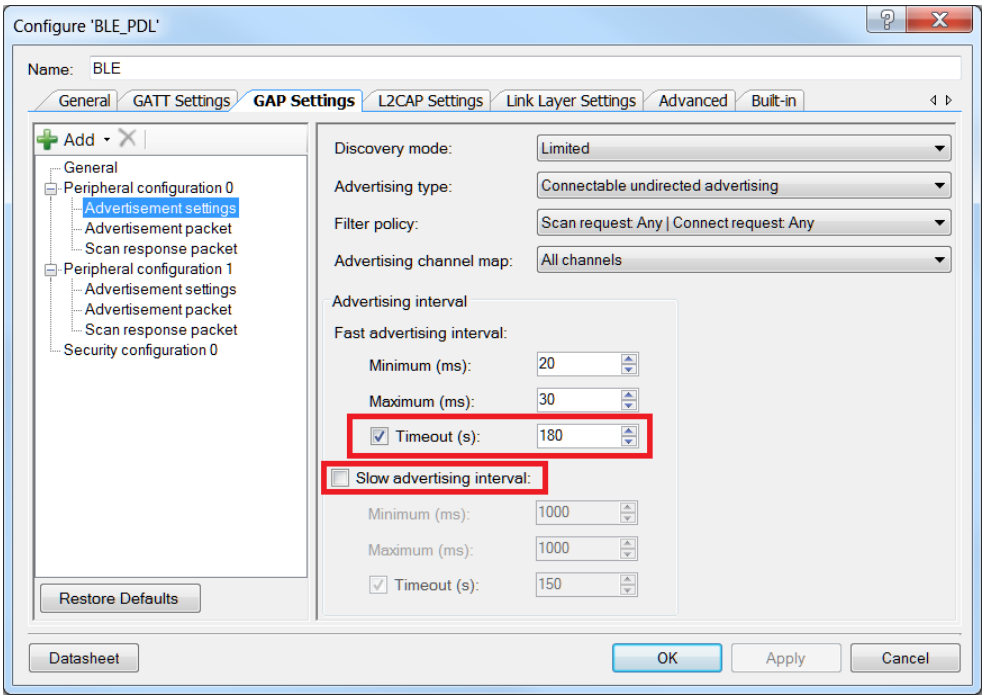


Figure 15. GAP Settings: Peripheral configuration 0 → Advertisement Packet

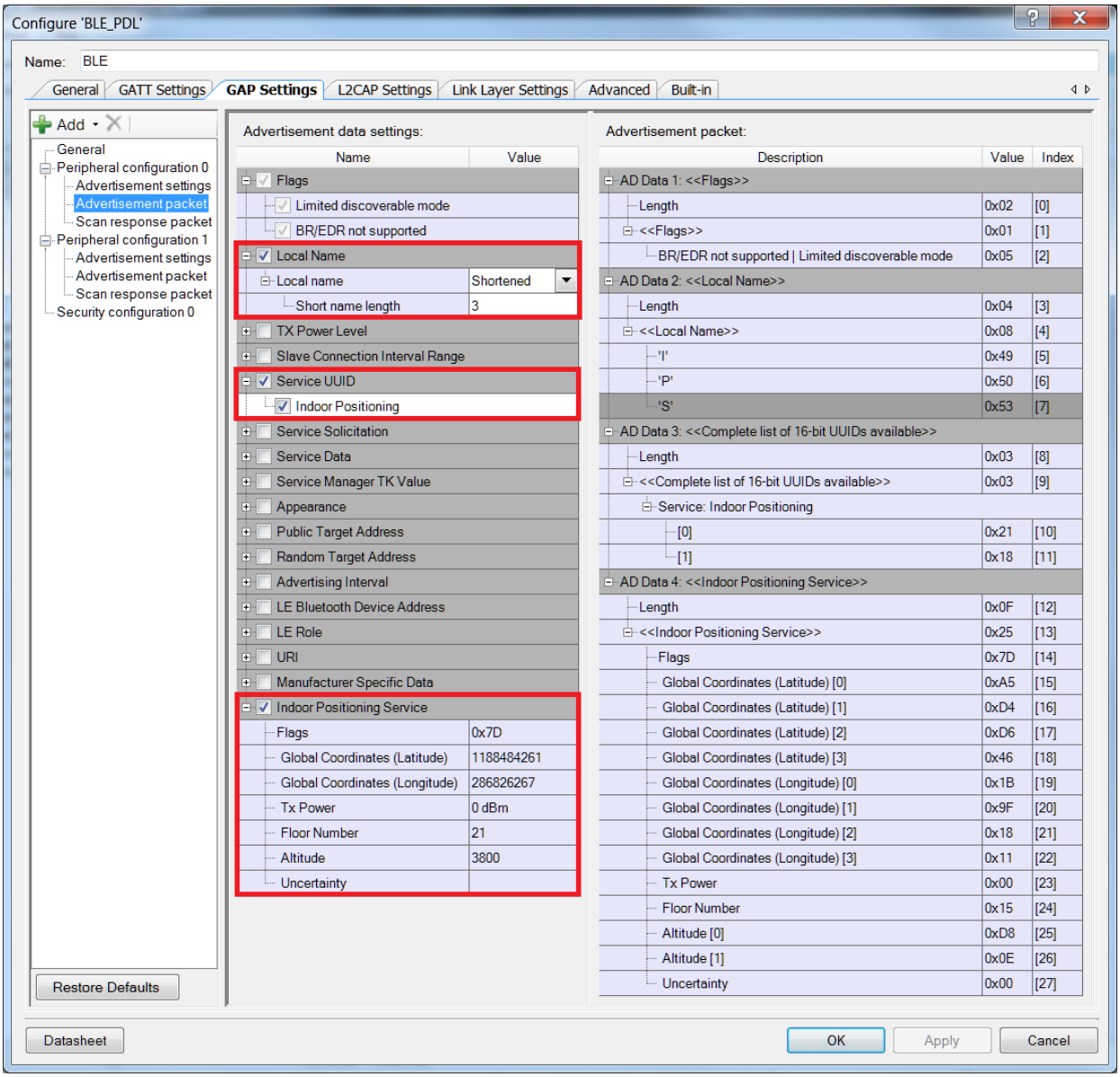


Figure 16. GAP Settings: Peripheral configuration 1 → Advertisement Settings

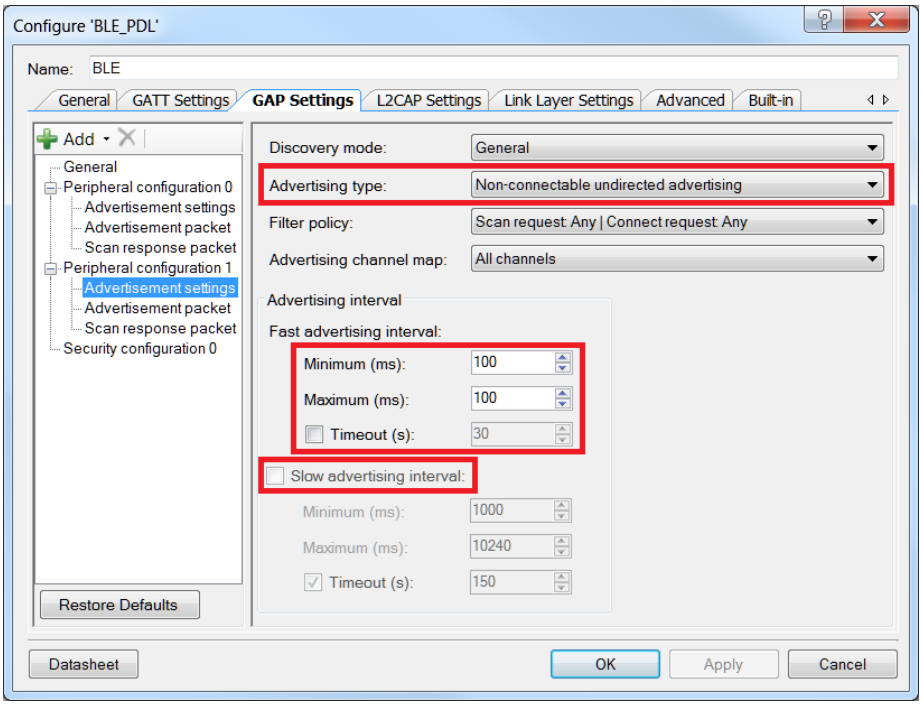


Figure 17. GAP Settings: Peripheral configuration 1 → Advertisement Packet

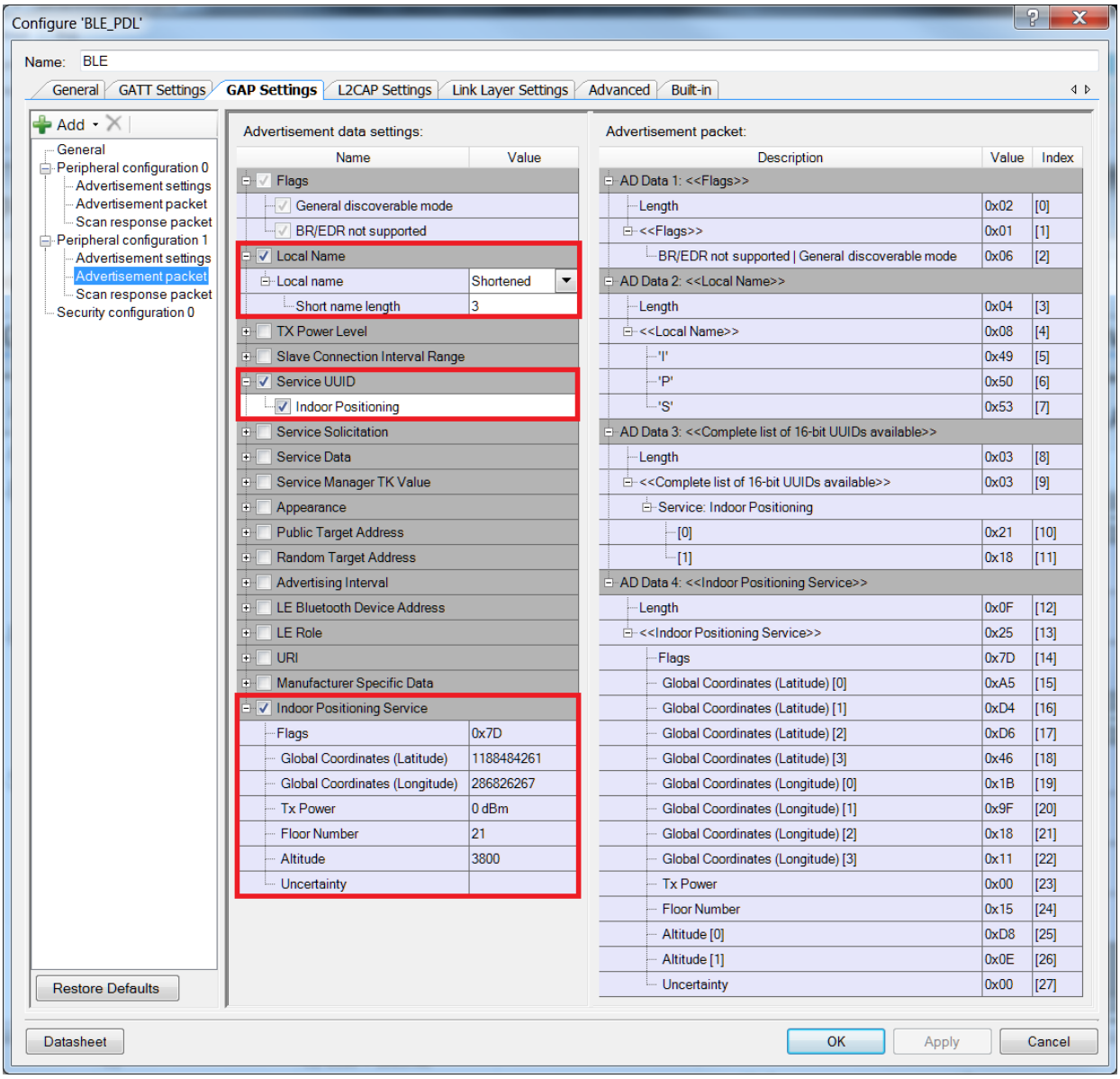
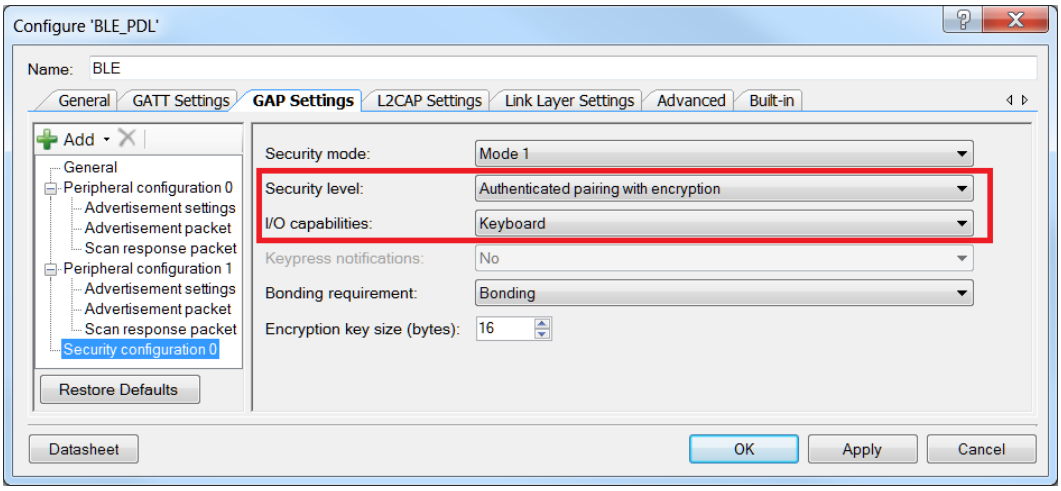


Figure 18. Security Settings



Switching the CPU Cores Usage

This section describes how to switch between different CPU cores usage (Single core / Dual core) in the BLE PDL examples.

The BLE component has the CPU Core parameter that defines the cores usage. It can take the following values:

- **Single core (Complete Component on CM0+)** – only CM0+ will be used.
- **Single core (Complete Component on CM4)** – only CM4 will be used.
- **Dual core (Controller on CM0+, Host and Profiles on CM4)** – CM0+ and CM4 will be used: CM0+ for the Controller and CM4 for the Host and Profiles.

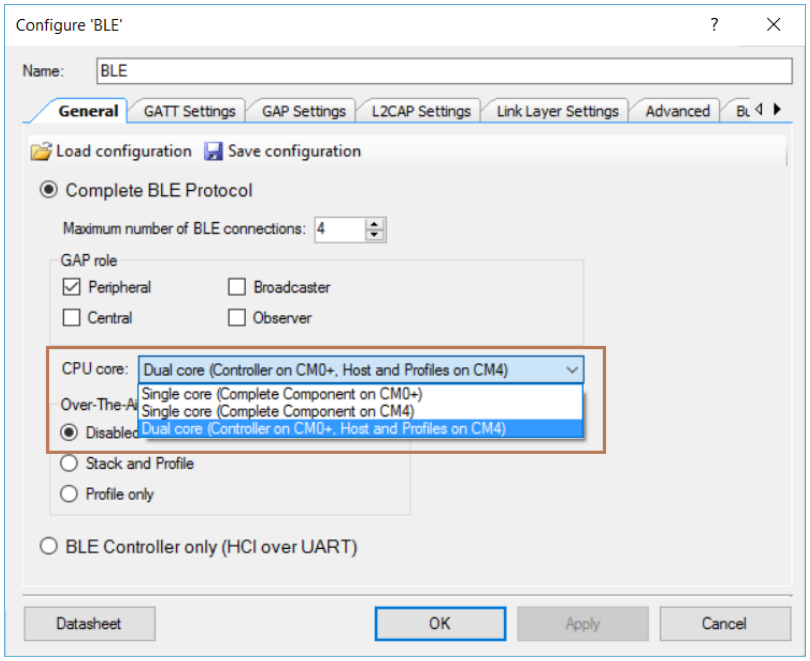
The BLE example structure allows easy switching between different CPU cores options. Important to remember:

- All application host-files must be run on the host core.
- The BLE subsystem (BLESS) interrupt must be assigned to the core where the controller runs.
- All additional interrupts (SW2, etc.) used in the example must be assigned to the host core.

Do the following to switch the CPU Cores usage:

1. In the BLE customizer **General** tab, select appropriate CPU core option.

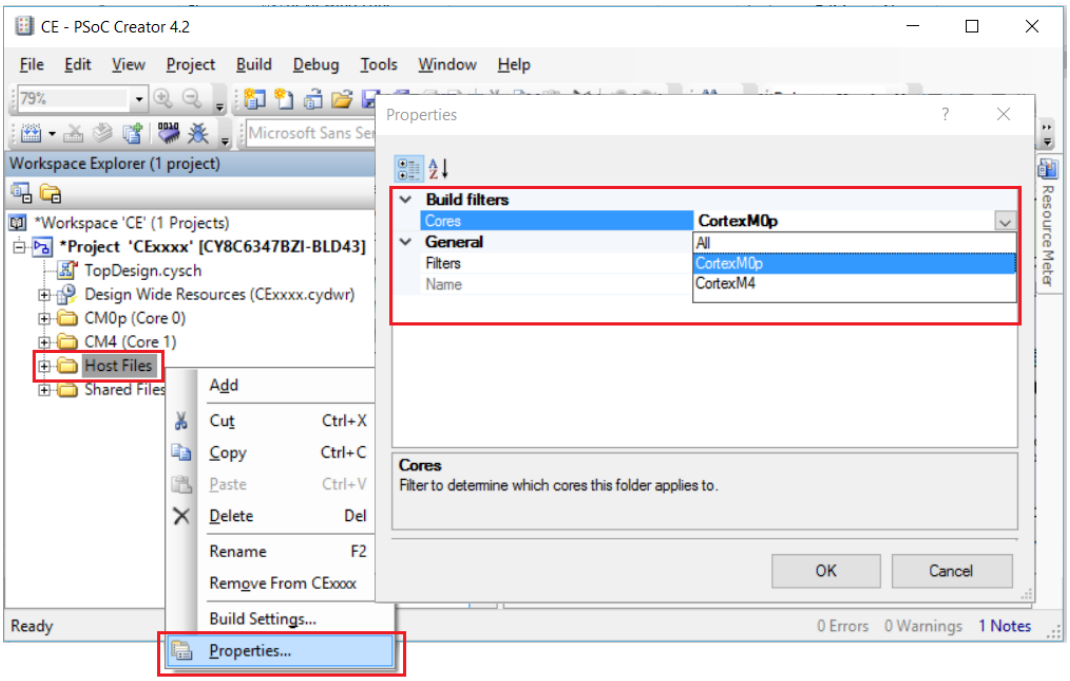
Figure 19. Select CPU Core



2. Identify the CPU on which host files will run. In the workspace explorer panel, right-click **Host Files**, choose **Properties**. Set the **Cores** property corresponding to the CPU core chosen in step 1, as shown in Figure 20.

   - For **Single core (Complete Component on CM0+)** option – CM0+
   - For **Single core (Complete Component on CM4)** option – CM4
   - For **Dual core (Controller on CM0+, Host and Profiles on CM4)** option – CM4

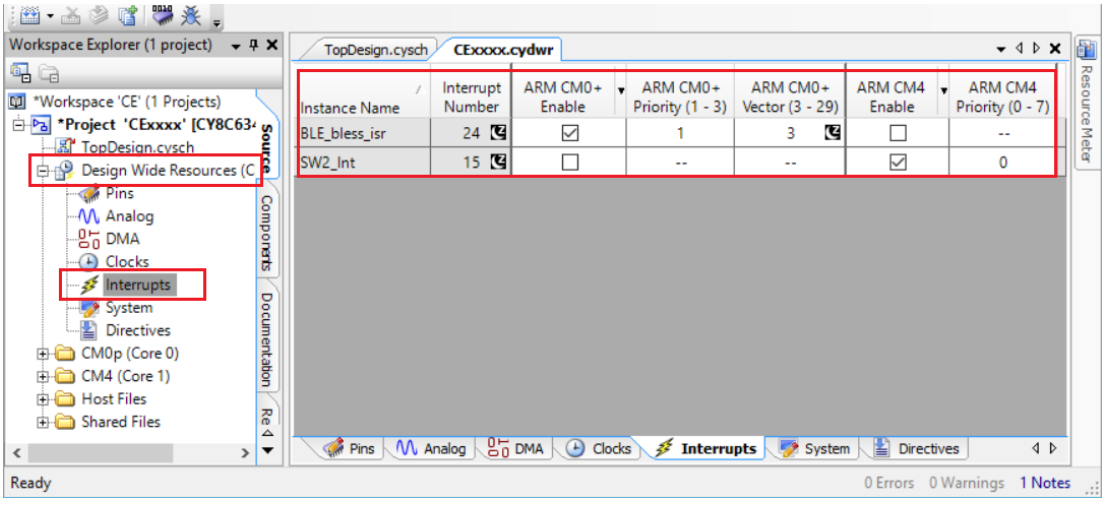
Figure 20. Change Core Properties



3. Assign BLE_bless_isr and other peripheral (button – SW2, timer(s), and so on) interrupts to the appropriate core in **DWR** > **Interrupts** tab:

   - For **Single core (Complete Component on CM0+)** option: BLE_bless_isr and peripheral interrupts on **CM0+**
   - For **Single core (Complete Component on CM4)** option: BLE_bless_isr and peripheral interrupts on **CM4**
   - For **Dual core (Controller on CM0+, Host and Profiles on CM4)** option: BLE_bless_isr interrupt on **CM0+,** other peripheral interrupts on **CM4**

Figure 21. Assign Interrupts



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.

Related Documents

Application Notes		
AN210781	Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity	Describes PSoC 6 BLE, and how to build a basic code example.
AN215656	PSoC 6 MCU Dual-CPU System Design	Presents the theory and design considerations related to this code example.
Software and Drivers		
CySmart – Bluetooth® LE Test and Debug Tool		CySmart is a Bluetooth® LE host emulation tool for Windows PCs. The tool provides an easy-to-use Graphical User Interface (GUI) to enable the user to test and debug their Bluetooth LE peripheral applications.
PSoC Creator Component Datasheets		
Bluetooth Low Energy (BLE_PDL) Component		The Bluetooth Low Energy (BLE_PDL) Component provides a comprehensive GUI-based configuration window to facilitate designing applications requiring BLE connectivity.
Device Documentation		
PSoC® 6 MCU: PSoC 63 with BLE. Datasheet.		PSoC® 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual
Development Kit (DVK) Documentation		
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit		

Document History

Document Title: CE217647 - BLE Indoor Positioning with PSoC 6 MCU with BLE Connectivity

Document Number: 002-17647

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6092398	NPAL	06/05/2018	New spec

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