CE215555 - BLE Multi Role with PSoC 6 MCU with BLE Connectivity

Objective

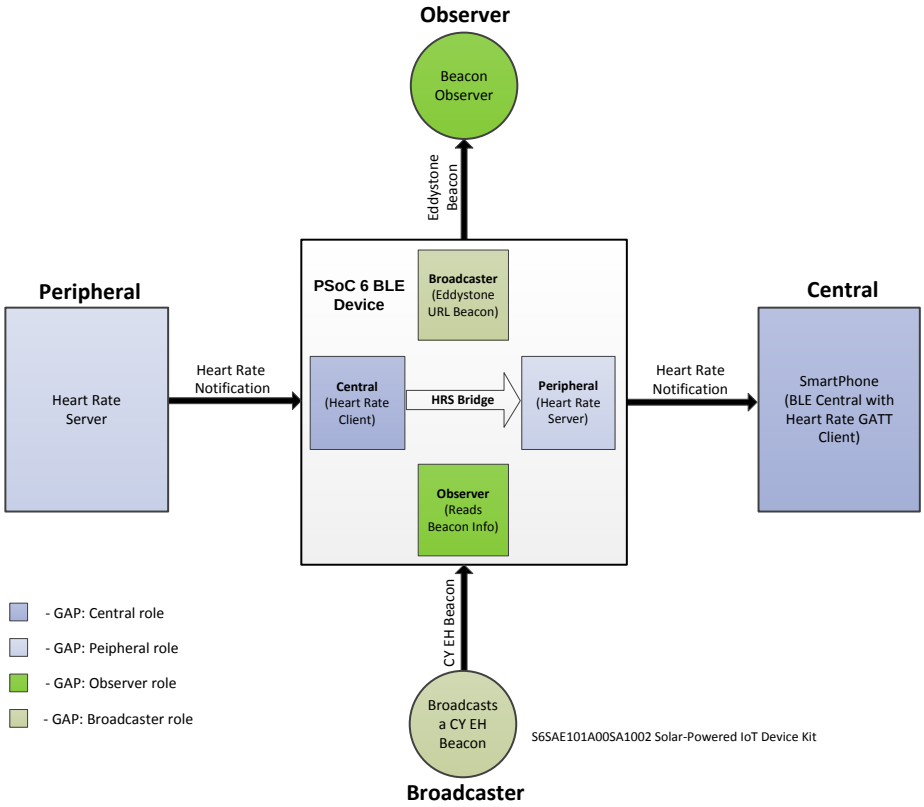
This example demonstrates the capability of the PSoC 6 BLE device to be in all Generic Access Profile (GAP) roles (Central, Peripheral, Observer, and Broadcaster) simultaneously.

Overview

This example demonstrates the device communication with the four peer Bluetooth Low Energy (BLE) devices, each operating in a different GAP role:

- The Central connection supports a Heart Rate Sensor (HRS) Generic Attribute Profile (GATT) Client and works with the existing HRS Sensor code example (CE217639) that runs on the PSoC 6 BLE device.
- The Peripheral connection supports the Heart Rate Sensor (HRS) GATT Server, which acts as a bridge to transfer data received from the peer Peripheral to the peer Central device. The peer Central device can be either a CySmart host emulation tool or Heart Rate Collector code example (CE217639) that runs on the PSoC 6 BLE device. This code example simulates heart-rate data and sends it to the peer Central device, or operates as a bridge for HRS data when a Peripheral device is connected.
- The Broadcaster profile continuously broadcasts the Eddystone URL beacon with Cypress web page.
- The Observer profile parses Cypress Energy Harvesting Beacon or AltBeacon and displays the beacon Universally Unique Identifier (UUID) and the Received Signal Strength Indication (RSSI) value.

Figure 1. Multi-Role Block Diagram



Requirements

Tool: PSoC Creator™ 4.2 or later

Programming Language: C (ARM® GCC 5.4-2016-q2-update or later)

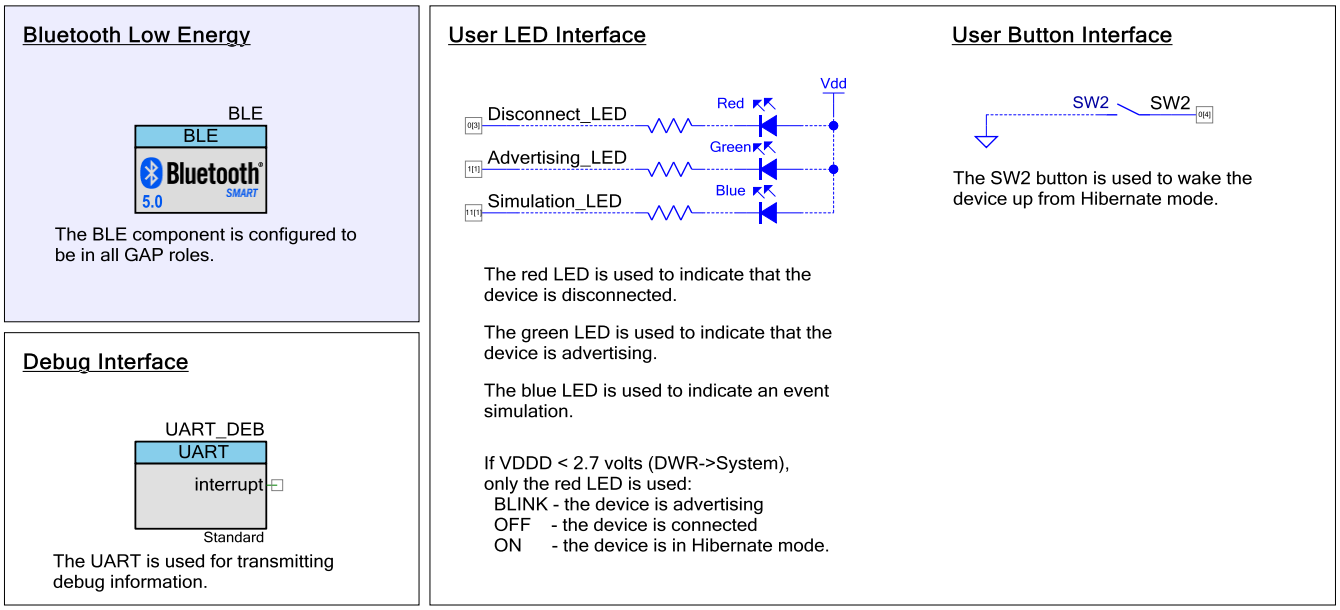
Associated Parts: All PSoC 6 parts

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

Design

Figure 2 shows the top design schematic.

Figure 2. BLE Multi-Role Code-Example Schematic



After a startup, the *main()* function initializes the BLE Component. To operate, the Component requires several callback functions to receive events from the BLE Stack. *AppCallBack()* receives general BLE events. *HeartRateCallBack()* and *BasCallBack()* receive events specific to the service's attribute operations. The example uses the UART Component for displaying debug information and entering commands through the Terminal emulator application. Table 1 lists the terminal commands.

Table 1. Terminal Commands List

Command	Description
a	Start connectable advertisement as a Heart Rate Monitor (HRM) Server.
s	Start scanning for an HRM sensor.
c	Send a connect request to the peer device (HRM sensor).
b	Start broadcasting as Eddystone URL Beacon.
o	Start scanning in the Observer role.
d	Send a disconnect request to the peer device.
p	Print the list of connected devices.
1	Stop scanning.
2	Stop advertising.

The above list is prompted to the Terminal emulator when you enter '**h**' in the application.

Design Considerations

Using UART for Debugging

Download and install a serial port communication program. Freeware such as Bray's Terminal and PuTTy are available on the web.

1. Connect the PC and kit with a USB cable.
2. Open the device manager program in your PC, find a COM port that the kit is connected to, and note the port number.
3. Open the serial port communication program and select the previously noted COM port.
4. Configure the Baud rate, Parity, Stop bits, and Flow control information in the Putty configuration window. The default settings: 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.
5. Start communicating with the device as explained in Operation.

UART debugging can be disabled by setting the `DEBUG_UART_ENABLED` to `DISABLED` in the *common.h* file.

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

The BLE examples' structure allows easy switching between different CPU cores options.

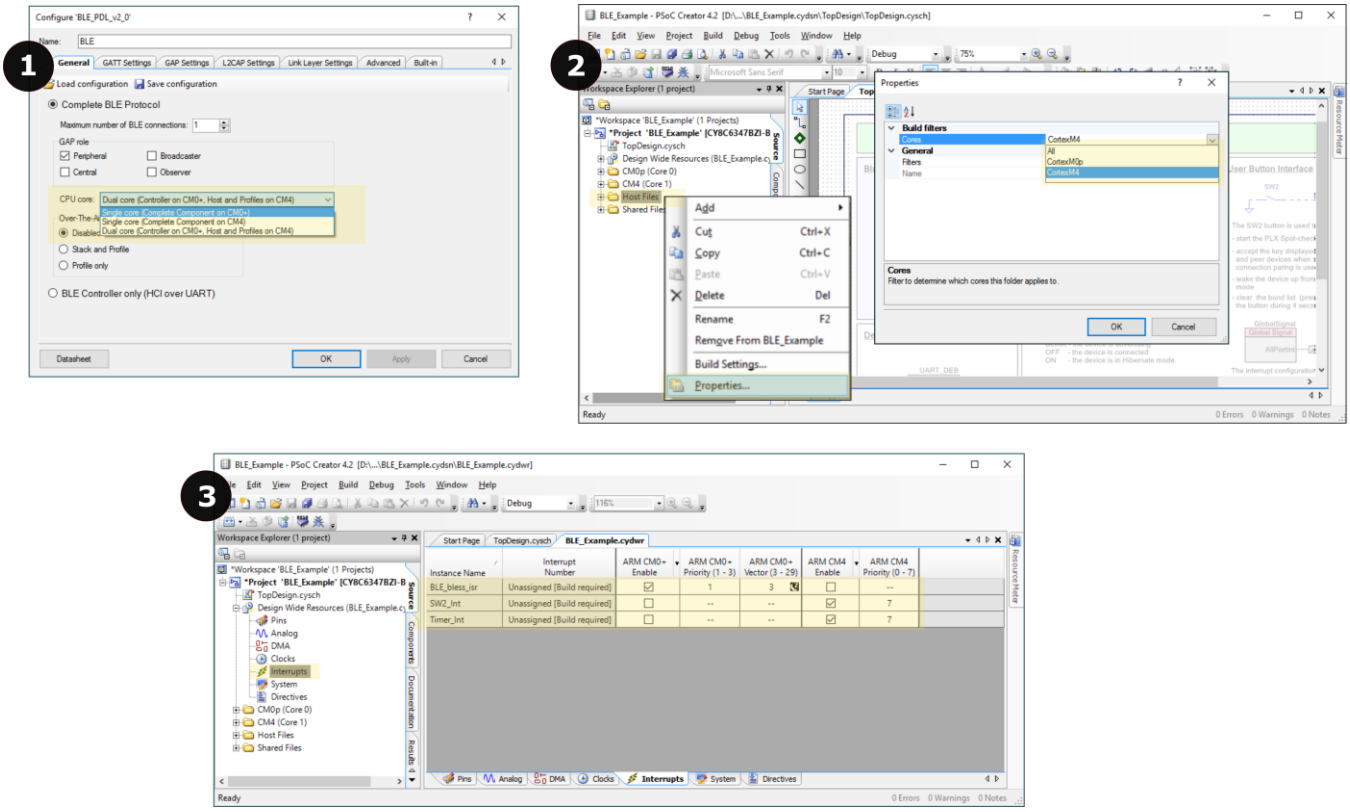
Important to remember:

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

Steps for switching the CPU Cores usage:

1. In the BLE customizer **General** tab, select appropriate CPU core option.
2. Change the cores Properties to CortexM4 or CortexC0p for the project folder Host Files in dependence of which CPU core was chosen in step 1. It should be:
   - 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**
3. Assign the BLE_bless_isr and other peripheral (button – SW2, timer(s) etc.) interrupts to 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 3. Steps for Switching the CPU Cores Usage



Hardware Setup

The code example was created for the CY8CKIT-062 PSoC® 6 BLE Pioneer Kit.

Table 2 lists the pin assignment and connections required on the development board for the 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]	
Advertising_LED	P1[1]	The green color of the RGB LED
Disconnect_LED	P0[3]	The red color of the RGB LED
Simulation_LED	P11[1]	The blue color of the RGB LED
SW2	P0[4]	

LED Behavior for V_{DDD} Voltage < 2.7 V

If the V_{DDD} voltage is set to less than 2.7 V in the DWR settings of the **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.

Table 3 lists the PSoC Creator components used in this example and the hardware resources used by each of the components.

Table 3. PSoC Creator Components List

Component	Hardware Resources
UART_DEB	1 SCB
BLE	1 BLE, 1 Interrupt
SW2	1 pin
Disconnect_LED, Advertising_LED, Simulation_LED	3 pins

Parameter Settings

The BLE Component is configured to support the GAP Peripheral, Central, Broadcaster, and Observer roles:

- In the GAP Central role, the component operates as a GATT Client with the Heart Rate Service (HRS), Battery Service (BAS) and Device Information Service (DIS).
- In the GAP Peripheral role, the component operates as a GATT Server with the same set of services and additional Custom Service with a 16-bit UUID equal to 0xFEAA (Eddystone -URL Service).
- In the GAP Broadcaster role, the component uses the Broadcaster configuration 0 with the following parameters:
  - Advertising type: Non-connectable undirected advertising
  - Advertising packet:
    - Service UUID: Eddystone -URL Service
    - Service Data: Eddystone -URL Service, Data: 0x10, 0x00, 0x00, ‘c’, ‘y’, ‘p’, ‘r’, ‘e’, ‘s’, ‘s’, 0x00
  - In the GAP Observer role, the component uses the Observer configuration 0 with the following parameters:
    - Scanning state: Passive
    - Fast scan parameters:
      - Scan window: 5 ms
      - Scan interval: 5 ms

The BLE Component is also configured to have:

- Public Device Address: 00A050-000201
- Device name: BLE Multi Role
- Security Level: Unauthenticated pairing with encryption
- Bonding requirements: Bonding

Figure 4. General Settings

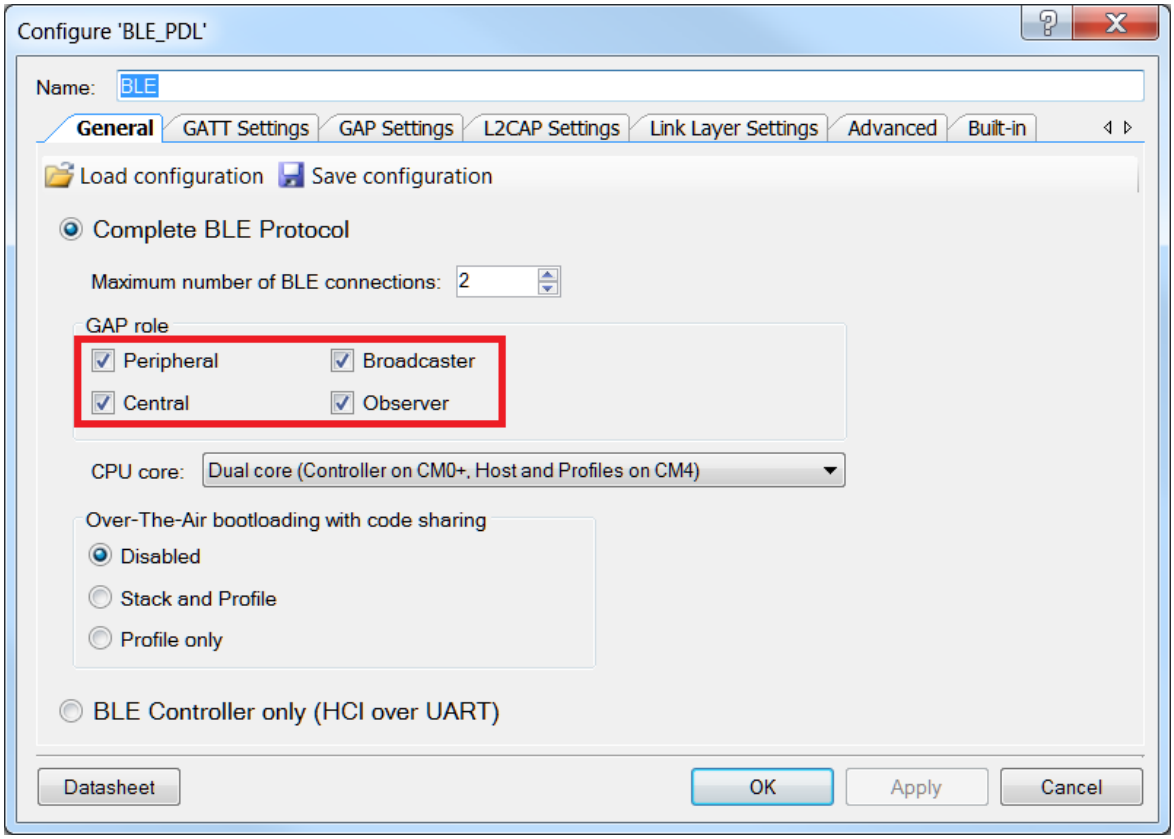


Figure 5. GATT Settings

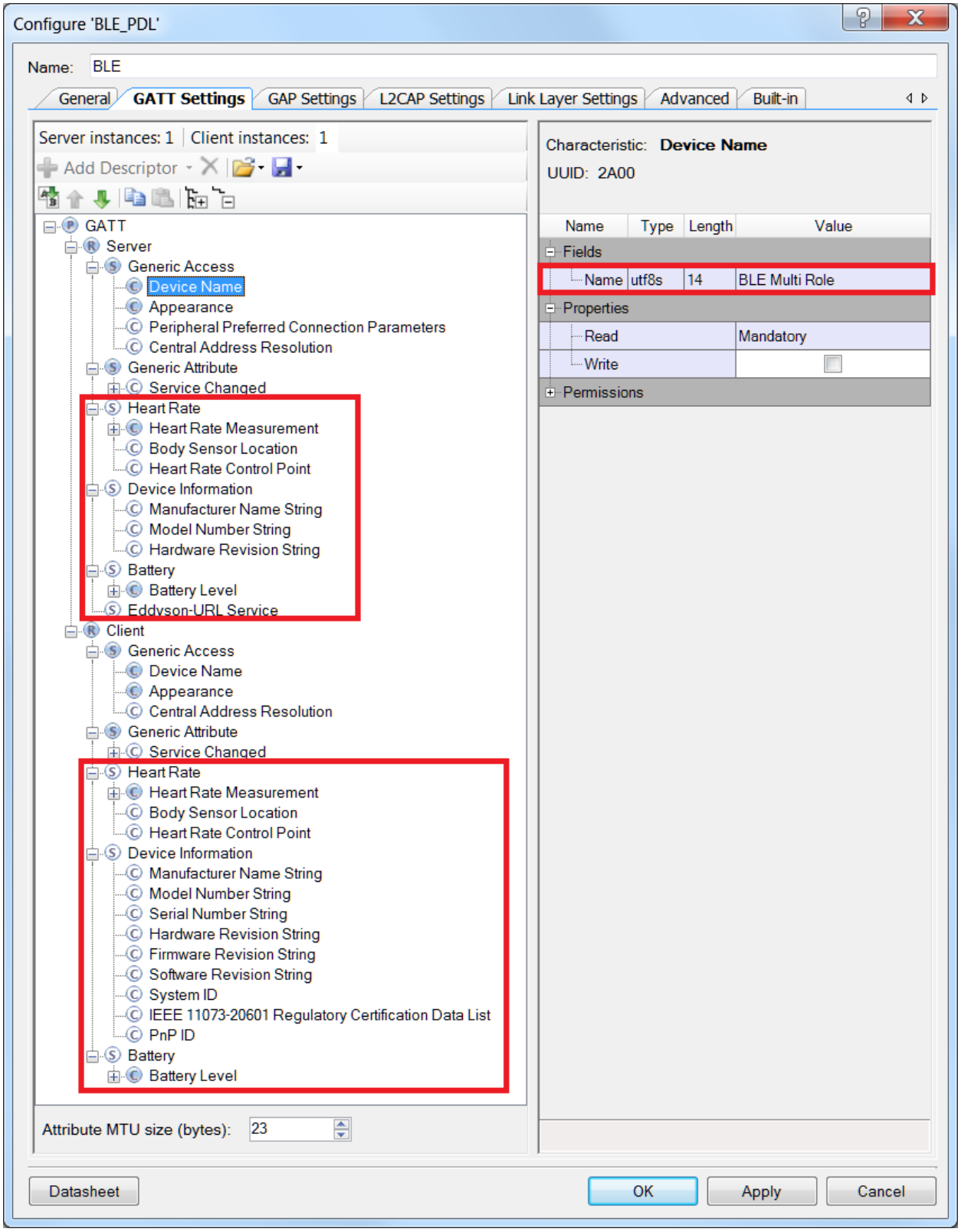


Figure 6. GAP Settings

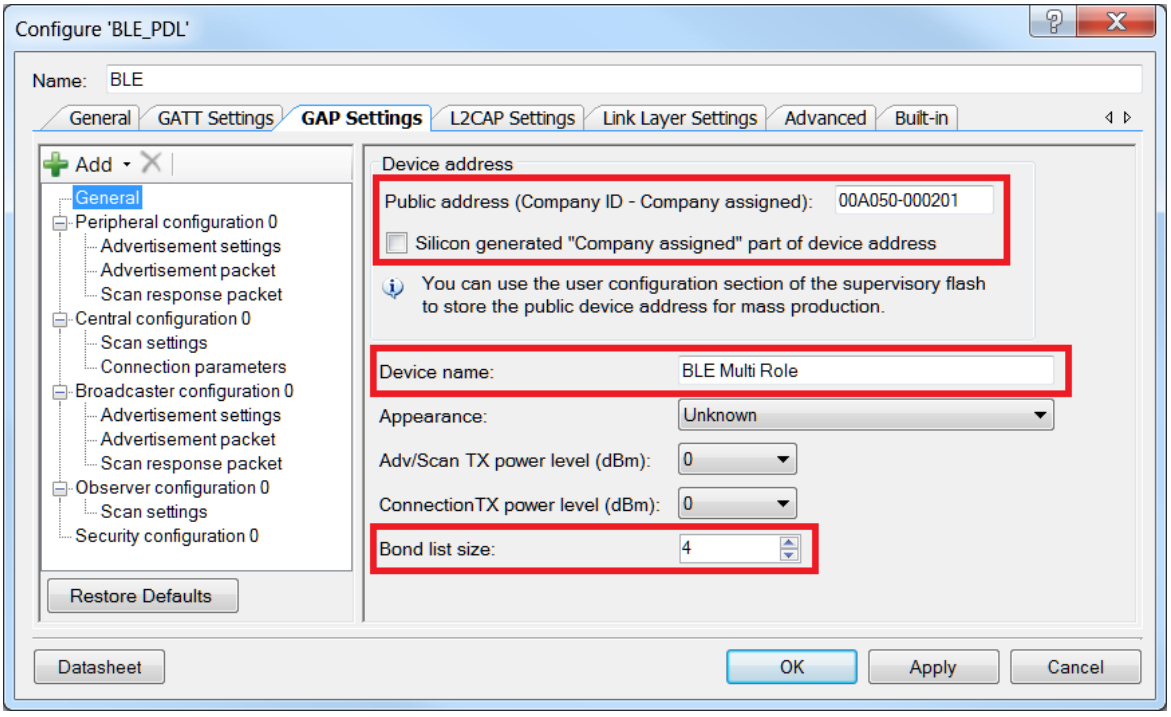


Figure 7. GAP Settings → Peripheral → Advertisement Settings

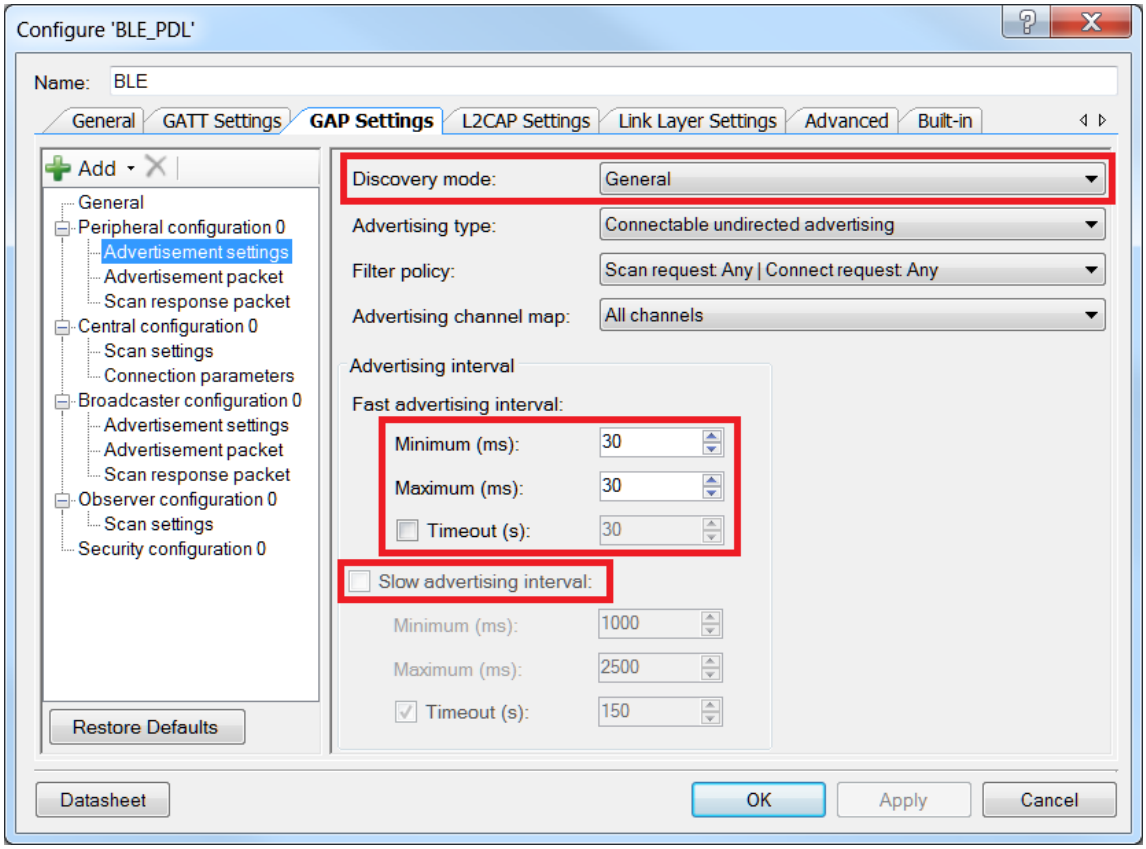


Figure 8. GAP Settings → Peripheral → Advertisement Packet

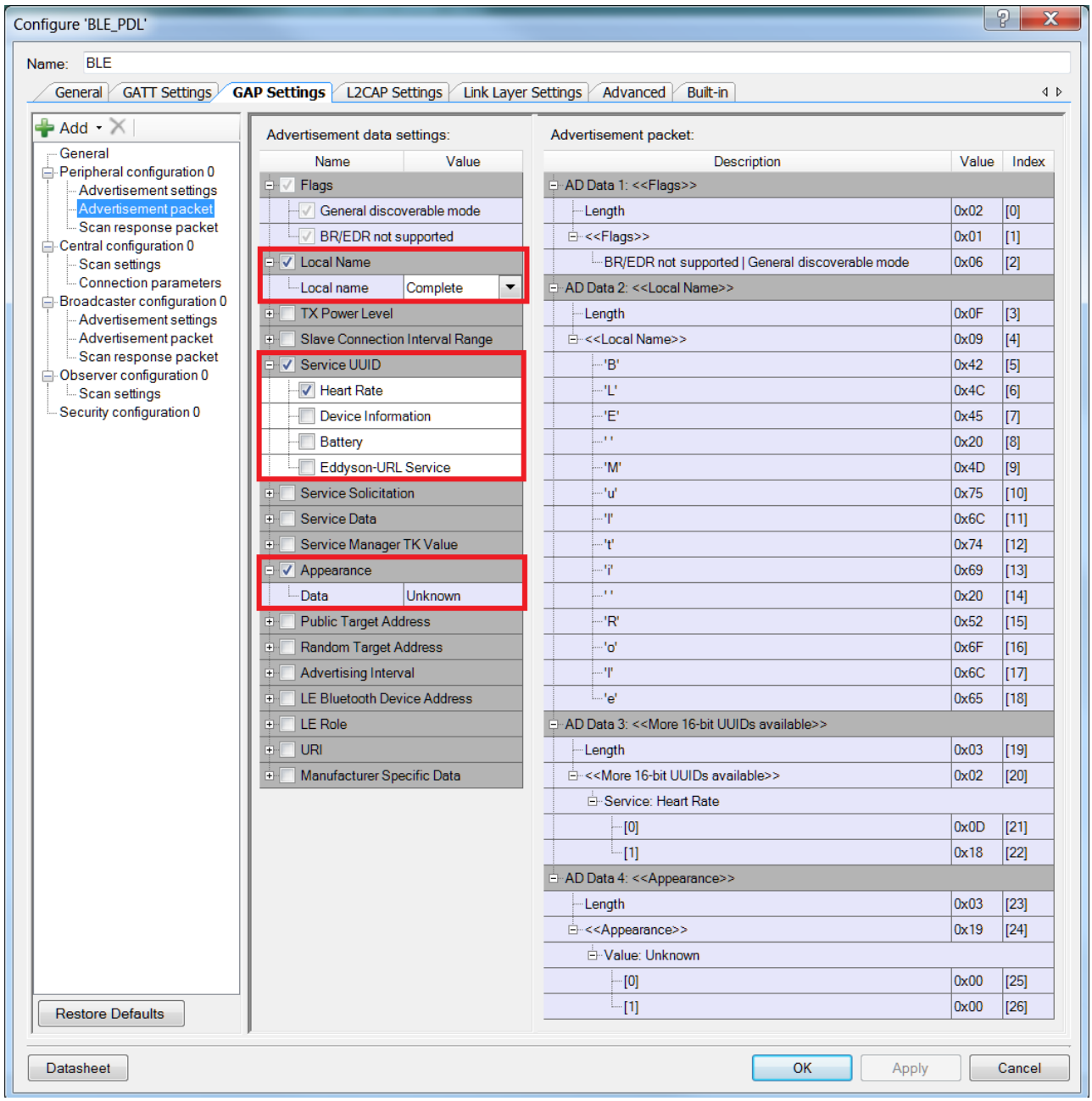


Figure 9. GAP Settings → Central →Scan Settings

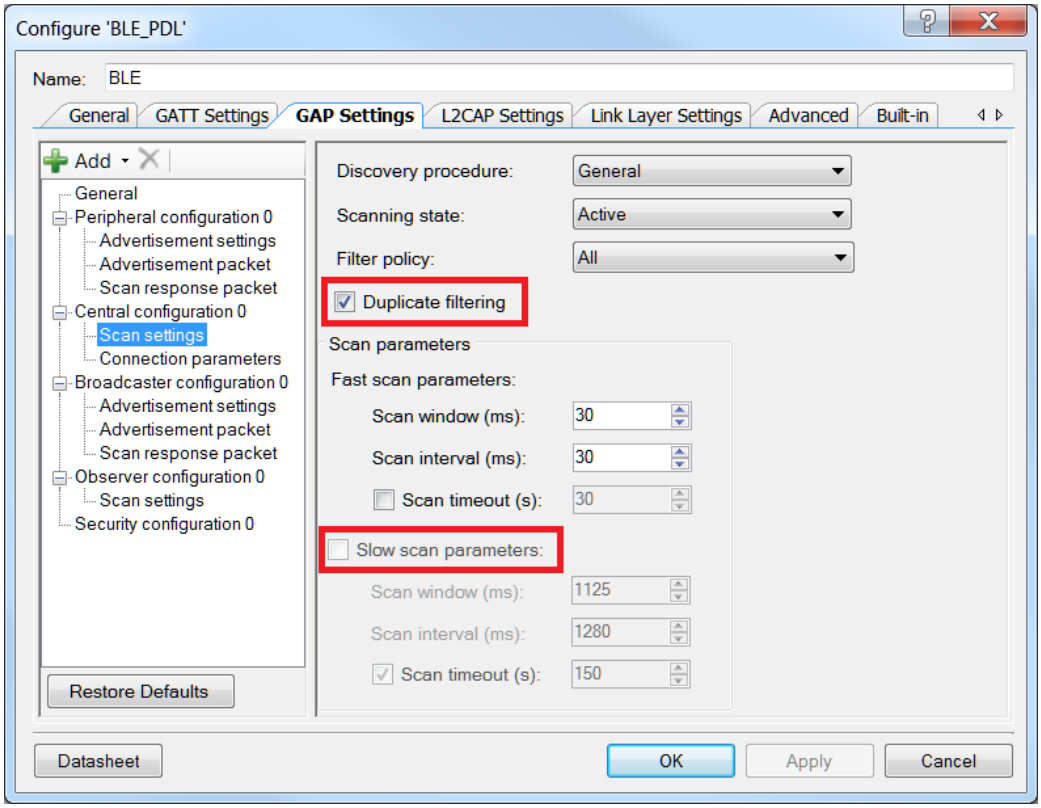


Figure 10. GAP Settings → Broadcaster → Advertisement Settings

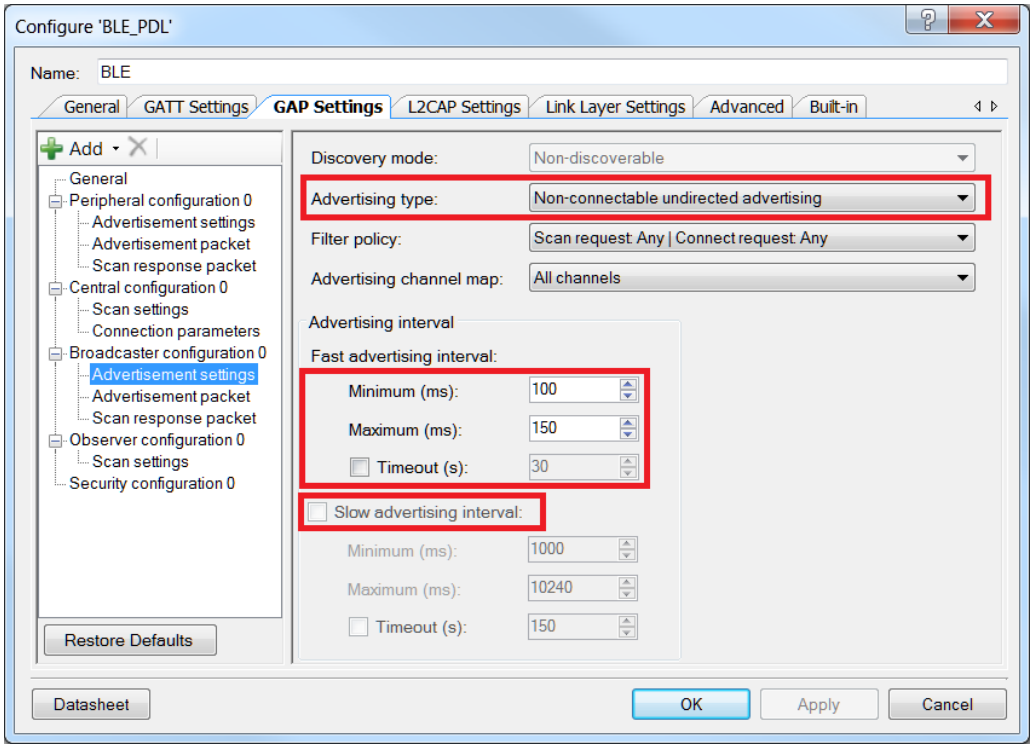


Figure 11. GAP Settings → Broadcaster → Advertisement Packet

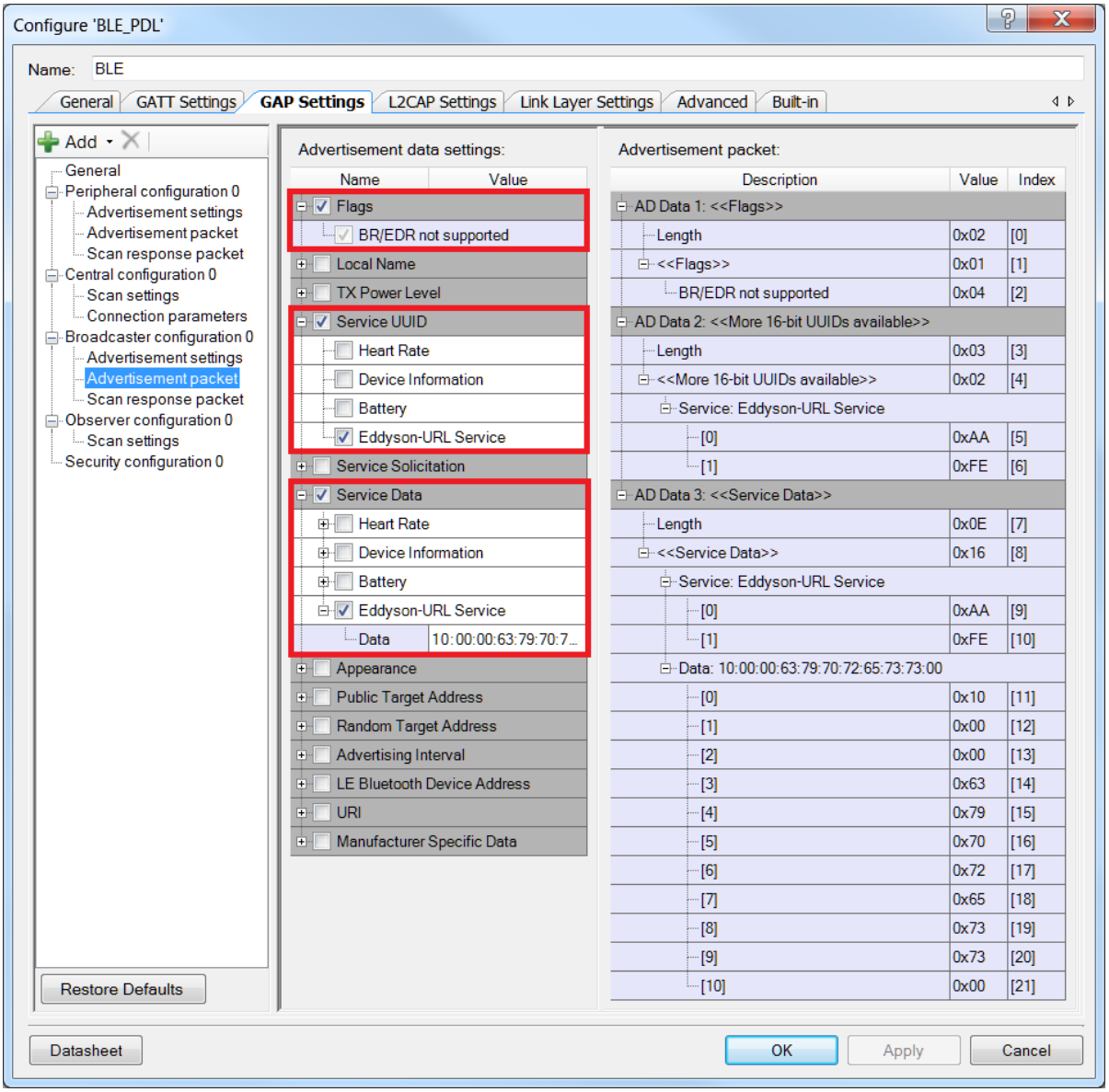


Figure 12. GAP Settings → Observer → Scan Settings

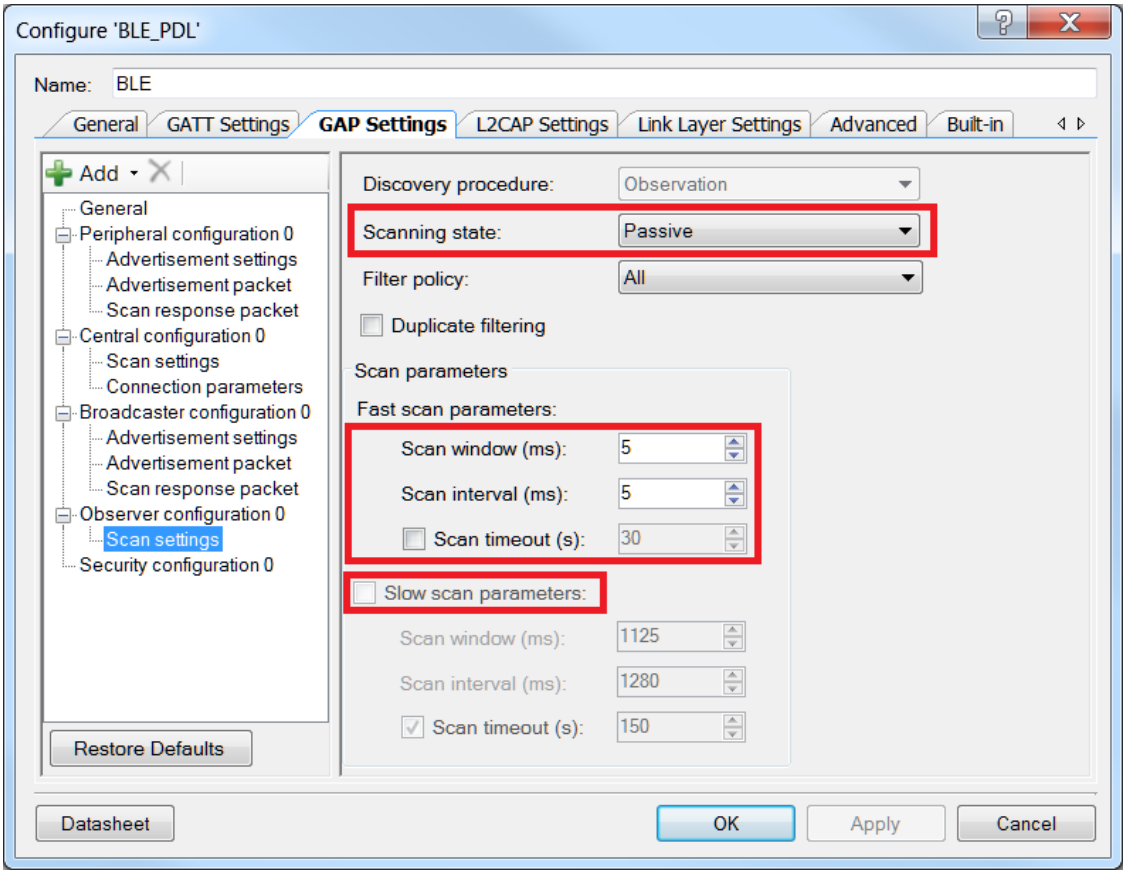
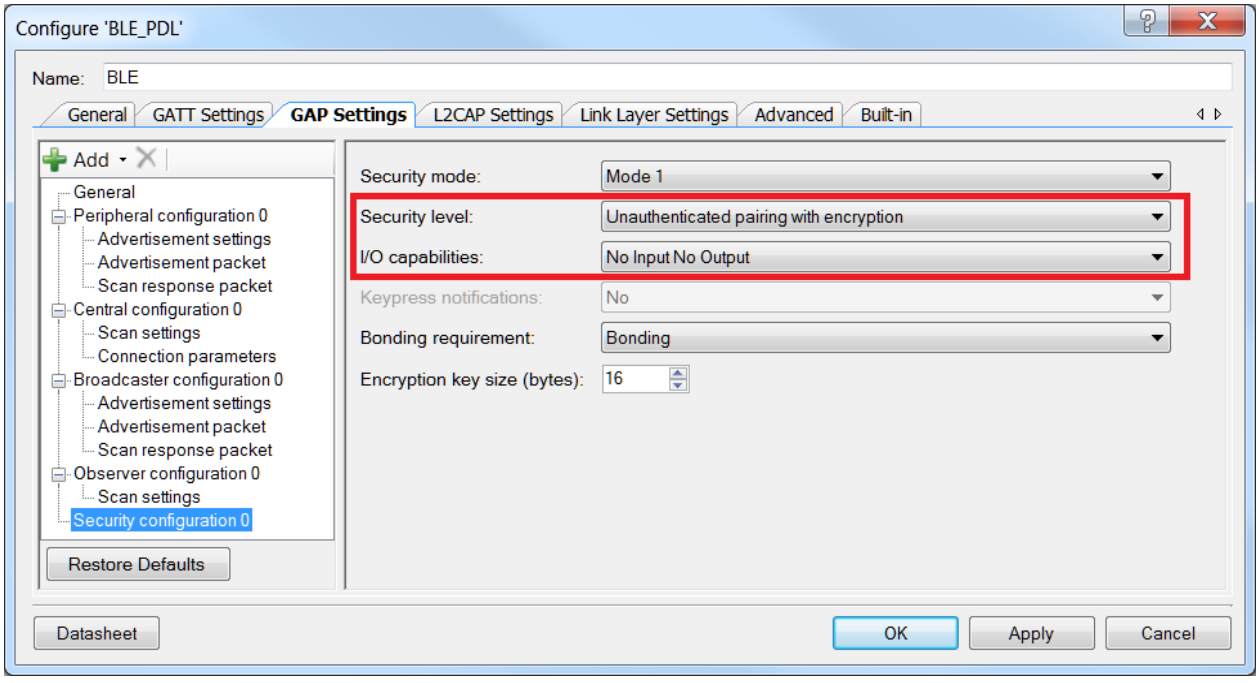


Figure 13. GAP Settings → Security Configuration



Operation

1. Build and program the BLE Multi-Role project into CY8CKIT-062 PSoC® 6 BLE Pioneer Kit.
2. Build and program the BLE HRS Sensor project (CE217639) into the CY8CKIT-062 PSoC® 6 BLE Pioneer Kit. The BLE HRS Sensor project has implemented simulation of all the constituents of the Heart Rate Measurement characteristic. The simulation includes the Heart Rate simulation, RR-Interval calculation, Energy Expended simulation, and simulation of rare sensor disconnection. The output of simulated data may appear in the debug serial terminal as follows:

   CY_BLE_EVT_HRSC_NOTIFICATION: **Heart Rate**: 156 **Energy Expended**: 11040 **RR-Interval 0**: 384 **RR-Interval** 1: 385

   CY_BLE_EVT_HRSC_NOTIFICATION: **Sensor Contact is supported but not detected**

   The message “Sensor Contact is supported but not detected” is received during simulation of the rare sensor disconnection: the Heart Rate Measurement characteristic with the Flag “Sensor Contacts Status Bits” is equal to “Sensor Contact feature is supported, but not detected”. For details on the Heart Rate Service characteristic data structures and behavior, see n the HRS Specification.

3. Run a serial port communication program (Bray’s Terminal, Putty, and so on) for the BLE Multi Role project.
4. Perform the Central operation. In the serial port communication program window, press ‘**s**’ to start scanning for advertising devices. When the scan report from the device with the Heart Rate Service in the advertisement data is received, the device address is automatically appended to the peerAddr list. Press ‘**c**’ to connect, and then select the number that corresponds to the device with the required address. To connect to another device, press ‘**c**’ again, and select a device for connection.

   The output on the client’s serial port communication program may appear as follows:

	BLE Multi Role Example BLE Stack Version: 5.0.0.718 CY_BLE_EVT_STACK_ON, StartAdvertisement Select operation: 'a' -- start connectable advertisement as HRM Server 's' -- start scanning for the HRM sensor 'c' -- send connect request to peer device (HRM sensor). 'b' -- start broadcasting as Eddystone-URL Beacon
Operation Menu	'o' -- start observer role 'd' -- send disconnect request to peer device 'p' -- print list of connected devices ----------
	'1' -- stop scanning '2' -- stop advertising CY_BLE_EVT_SET_DEVICE_ADDR_COMPLETE CY_BLE_EVT_LE_SET_EVENT_MASK_COMPLETE CY_BLE_EVT_GET_DEVICE_ADDR_COMPLETE: 00a050000201 CY_BLE_EVT_SET_TX_PWR_COMPLETE CY_BLE_EVT_SET_TX_PWR_COMPLETE CY_BLE_EVT_GAPP_ADVERTISEMENT_START_STOP, state: 2 CY_BLE_EVT_GAP_KEYS_GEN_COMPLETE
Press 's'	**s** **Start scanning for the HRM sensor** CY_BLE_GapcStartScan API Success CY_BLE_EVT_GAPC_SCAN_START_STOP GAPC_START_SCANNING DV type: 0x0 address: 6240cc5f37e4, rssi - -78 dBm, data - 02 01 06 **----------** uuid: **HEART RATE SERVICE - YES**, added to the connect list ADV type: 0x0 address: 00a050000201, rssi - -73 dBm, data - 02 01 06 0f 09 42 4c 45 20 4d 75 6c 74 69 20 52 6f 6c 65 03 02 0d 18 03 19 00 00 **----------** ADV type: 0x0 address: 49bc48f75324, rssi - -79 dBm, data - 02 01 06 ADV type: 0x0 address: 00a05060544e, rssi - -85 dBm, data - 02 01 06 0c 09 43 59 38 43 4b 49
Press 'c'	**c** **Detected device:** Device 1: 00a050000201 select device for connection: (1..1):
Press '1'	**1**

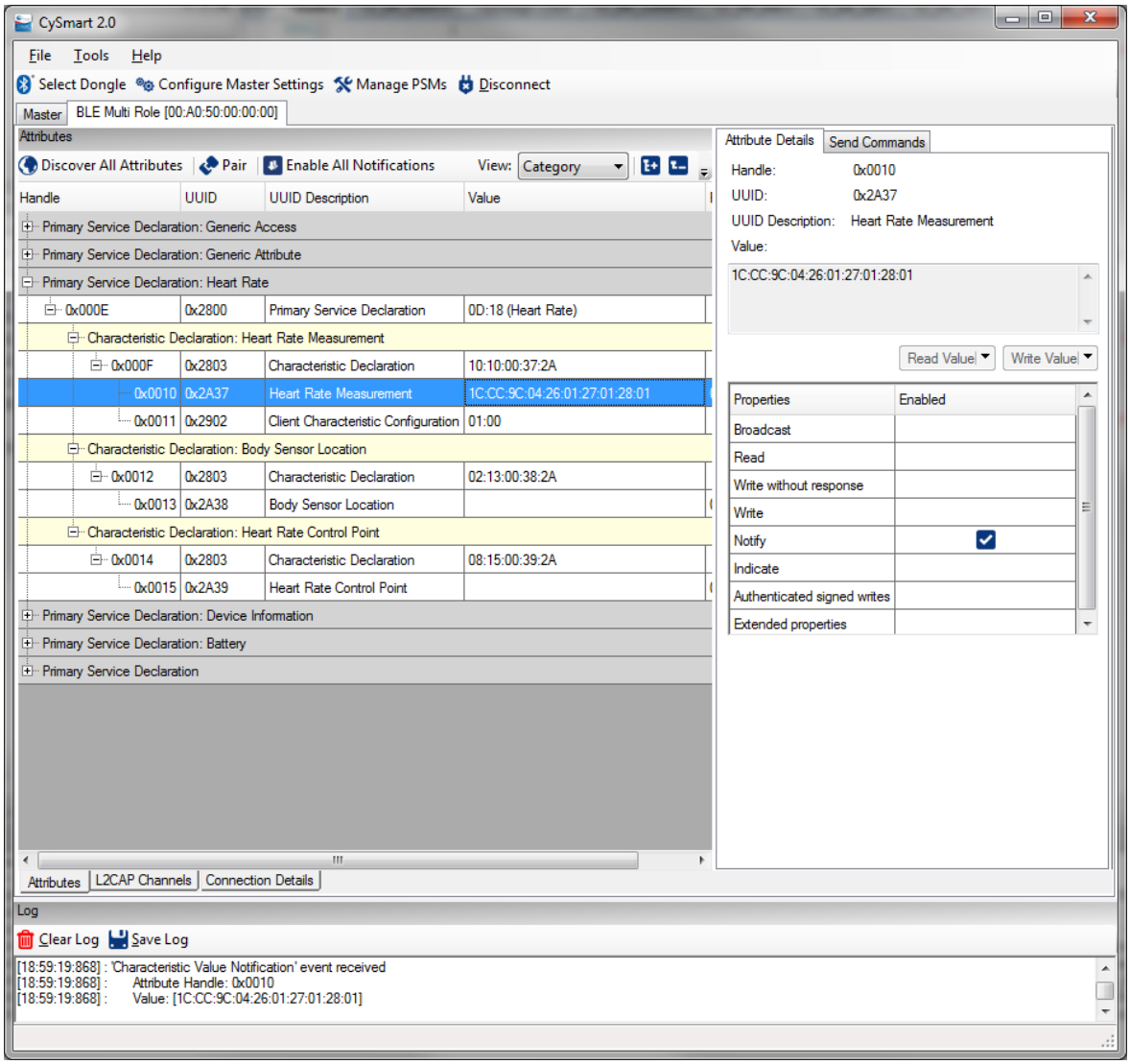
Service Discovery	**Connecting to the device (address - 00a050000201)** CY_BLE_EVT_GATT_CONNECT_IND: attId 3, bdHandle 7 CY_BLE_EVT_GAP_DEVICE_CONNECTED CY_BLE_EVT_GAPC_SCAN_START_STOP Scan complete CY_BLE_EVT_GAP_SMP_NEGOTIATED_AUTH_INFO: bdHandle=7, security=1, bonding=1, ekeySize=10, err=0 CY_BLE_EVT_GAP_ENCRYPT_CHANGE: 0 CY_BLE_EVT_GAP_KEYINFO_EXCHNGE_CMPLT CY_BLE_EVT_GAP_AUTH_COMPLETE: security:1, bonding:1, ekeySize:10, authErr 0 Cy_BLE_GATTC_StartDiscovery CY_BLE_EVT_STACK_BUSY_STATUS: 1 CY_BLE_EVT_PENDING_FLASH_WRITE CY_BLE_EVT_STACK_BUSY_STATUS: 0 Discovery is complete: attId=3, bdHandle=7 service with UUID 0x1800 has range from 0x1 to 0x9 service with UUID 0x1801 has range from 0xa to 0xd service with UUID 0x180f has range from 0x1d to 0x21 service with UUID 0x180a has range from 0x16 to 0x1c service with UUID 0x180d has range from 0xe to 0x15
Notifications	Enable notifications Cy_BLE_HRSC_SetCharacteristicDescriptor() successful. CY_BLE_EVT_HRSC_WRITE_DESCR_RESPONSE CY_BLE_EVT_HRSC_NOTIFICATION: Heart Rate: 84 EnergyExpended: 0 RR-Interval 0: 833 RR-Interval 1: 714 CY_BLE_EVT_HRSC_NOTIFICATION: Heart Rate: 96 EnergyExpended: 0 RR-Interval 0: 625 CY_BLE_EVT_HRSC_NOTIFICATION: Heart Rate: 108 EnergyExpended: 0 RR-Interval 0: 555 CY_BLE_EVT_HRSC_NOTIFICATION: Heart Rate: 120 EnergyExpended: 0 RR-Interval 0: 500 RR-Interval 1: 501 CY_BLE_EVT_HRSC_NOTIFICATION: Sensor Contact is supported but not detected CY_BLE_EVT_HRSC_NOTIFICATION: Sensor Contact is supported but not detected CY_BLE_EVT_HRSC_NOTIFICATION: Sensor Contact is supported but not detected CY_BLE_EVT_HRSC_NOTIFICATION: Heart Rate: 168 EnergyExpended: 0 RR-Interval 0: 357 RR-Interval 1: 358 CY_BLE_EVT_HRSC_NOTIFICATION: Heart Rate: 180 EnergyExpended: 0 RR-Interval 0: 333 RR-Interval 1: 334 RR-Interval 2: 335

5. Peripheral operation. Launch the CySmart Central Emulation tool and select the connected dongle in a dialog window.
   - Click **Start Scan** to discover the available devices.
   - Select **BLE Multi Role** from the list of the available devices and connect to it.
   - Select **Yes** to a pairing request received from the peer device. The output may appear as follows (press '**p**' to check that the server is connected).

Slave Connection Log	CY_BLE_EVT_GATT_CONNECT_IND: attId 2, bdHandle 12 CY_BLE_EVT_GAP_DEVICE_CONNECTED CY_BLE_GapAuthReq API Success 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=12, security=3, bonding=1, ekeySize=10, err=0 CY_BLE_EVT_GAP_ENCRYPT_CHANGE: 0 CY_BLE_EVT_GAP_AUTH_COMPLETE: security:1, bonding:1, ekeySize:10, authErr 0 CY_BLE_EVT_PENDING_FLASH_WRITE Store bonding data, status: 13, pending: 1 Store bonding data, status: 0, pending: 0 CY_BLE_EVT_GATTS_READ_CHAR_VAL_ACCESS_REQ: handle: d Heart Rate Measurement Notification is Enabled Store bonding data, status: 0, pending: 0 CY_BLE_EVT_GATTS_READ_CHAR_VAL_ACCESS_REQ: handle: 11
Press 'p'	**p** --- Connected devices list: 1. address: 00a050000201 bdHandle:13 attId:3 CLIENT 2. address: 00a05012c20e bdHandle:12 attId:2 SERVER ---

- Click **Discover All Attributes**, then **Enable All Notifications**. Observe the received characteristic values as shown in Figure 14.

Figure 14. CySmart Central Emulation Tool (Windows OS)



6. Disconnect the device. Press '**d**' in the serial port communication program to disconnect the device.

Press 'd' Select Device for Disconnection Press '2' Disconnect SERVER Press 'd' Select Device for Disconnection Press '1' Disconnect CLIENT	**d** --- Connected devices list: 1. address: 00a050000201bdHandle:13 attId:3 CLIENT 2. address: 00a05012c20e bdHandle:12 attId:2 SERVER --- Select a device for disconnection: (1..2): 2 Selected device for disconnection: 2 CY_BLE_GapDisconnect param: bdHandle:12, reason:13 CY_BLE_GapDisconnect API Success CY_BLE_EVT_GATT_DISCONNECT_IND: attId=2, bdHandle=12 Disconnect peripheral StartAdvertisement start CY_BLE_EVT_GAP_DEVICE_DISCONNECTED, reason: 0 **d** --- Connected devices list: 1. address: 00a050000201 bdHandle:13 attId:3 CLIENT --- Select a device for disconnection: (1..1): 1 Selected device for disconnection: 1 CY_BLE_GapDisconnect param: bdHandle:13, reason:13 CY_BLE_GapDisconnect API Success CY_BLE_EVT_GATT_DISCONNECT_IND: attId=3, bdHandle=13 Disconnect Central Start broadcasting as Eddystone-URL Beacon with link to: http://www.cypress.com CY_BLE_EVT_GAPP_ADVERTISEMENT_START_STOP, state: 2

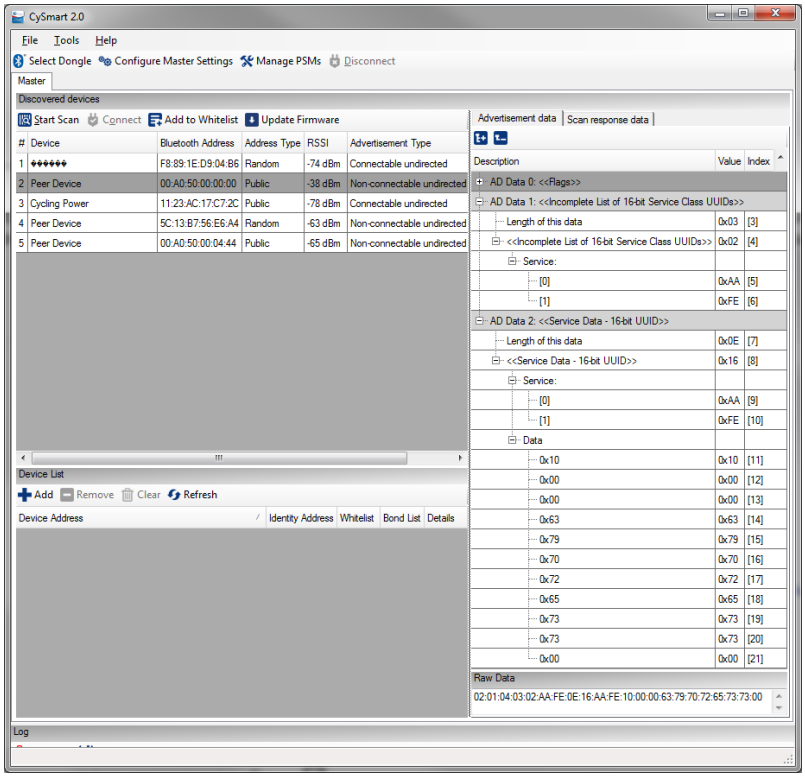
7. Perform the Broadcaster operation.
   - Press '**b**' to start broadcasting as Eddystone-URL Beacon with link to: http://www.cypress.com. An output on the client may appear as follows:

Press 'b'	Start broadcasting as Eddystone -URL Beacon with link to: http://www.cypress.com CY_BLE_EVT_GAPP_ADVERTISEMENT_START_STOP, state: 2

- Install the beacon scanner application on your Android/iPhone device (for example, iBeacon & Eddystone Scanner)
- Scan the beacons to find the required device as shown in Figure 15.

Figure 15. iBeacon & Eddystone Scanner (Android OS)

Figure 16. Broadcast Packet in CySmart



8. Perform the Observer operation:
   - Set up the S6SAE101A00SA1002 Solar-Powered IoT Device Kit or other Cypress's Energy harvesting Beacon as a BLE Beacon.
   - Press '**o**' to start scanning in Observer mode. An output on the client may appear as follows:

Press 'o'	Start scanning in Observer mode CY_BLE_GapcStartScan API Success CY_BLE_EVT_GAPC_SCAN_START_STOP GAPC_START_SCANNING CY_BLE_EVT_GAPC_SCAN_PROGRESS_RESULT: **ADDRESS**: 5d:47:ae:22:a3:a5, **RSSI**: -75 dBm, **data**: 02 01 04 1a ff 31 01 02 15 00 05 00 01 00 00 10 00 80 00 00 80 5f 9b 01 31 00 01 00 01 c3 **COMPANY ID**: 0x131, **RSSI_1M**: -61, **UUID**: 00 05 00 01 00 00 10 00 80 00 00 80 5f 9b 01 31 …..

Related Documents

Table 4 lists all relevant application notes, code examples, knowledge base articles, device datasheets, and Component datasheets.

Table 4. Related Documents

Application Notes		
AN210781	Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity	Describes the PSoC 6 MCU with BLE Connectivity, and how to build a basic code example.
AN215656	PSoC 6 MCU Dual-Core CPU System Design	Presents the theory and design considerations related to this code example.
Software and Drivers		
CySmart – BLE Test and Debug Tool		CySmart is a BLE host emulation tool for Windows PCs. The tool provides an easy-to-use GUI to enable the user to test and debug their BLE 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 Programmable System-on-Chip		PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual (TRM)
Development Kit (DVK) Documentation		
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit		

Document History

Document Title: CE215555 - BLE Multi Role with PSoC 6 MCU with BLE Connectivity

Document Number: 002-15555

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5968183	NPAL	11/15/2017	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

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

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