CE218044 - BLE Pulse Oximeter with PSoC 6 MCU with BLE Connectivity

Objective

This example demonstrates how to configure and use the Bluetooth Low Energy (BLE) Component APIs and application layer callbacks for the Pulse Oximeter Profile (PLXP).

Overview

This example demonstrates the core functionality of the BLE Component configured as a Pulse Oximeter Service (PLXS) device (GATT Server). The example simulates the PLX Spot-check Measurement and PLX Continuous Measurement characteristics. To conserve power, the device switches to Deep Sleep mode between the BLE connection intervals. Additionally, this project implements the following services as per the Pulse Oximeter Profile specification: BMS, BAS, DIS, and CTS.

Requirements

Tool: PSoC Creator™ 4.2 or later

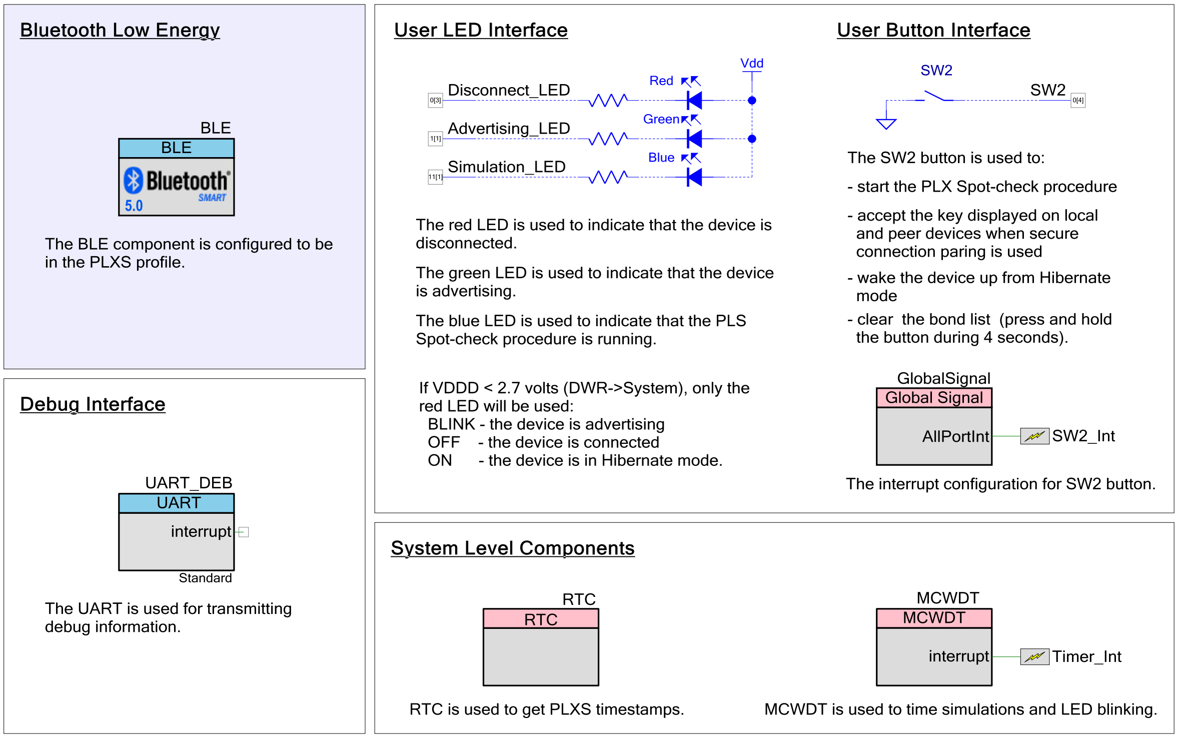
Programming Language: C (Arm® GCC 5.4-2016-q2-update or later)

Associated Parts: All PSoC® 6 MCU with Bluetooth Low Energy (BLE) Connectivity (PSoC 6 BLE) parts

Related Hardware: CY8CKIT-062 PSoC 6 BLE Pioneer Kit

Design

Figure 1. BLE Pulse Oximeter Code Example Schematic



The BLE component is configured to be in the PLXS profile.

The UART is used for transmitting debug information.

The red LED is used to indicate that the device is disconnected.

The green LED is used to indicate that the device is advertising.

The blue LED is used to indicate that the PLS Spot-check procedure is running.

If VDDD < 2.7 volts (DWR->System), only the red LED will be used:
BLINK - the device is advertising
OFF - the device is connected
ON - the device is in Hibernate mode.

The SW2 button is used to:

- start the PLX Spot-check procedure
- accept the key displayed on local and peer devices when secure connection paring is used
- wake the device up from Hibernate mode
- clear the bond list (press and hold the button during 4 seconds).

The interrupt configuration for SW2 button.

RTC is used to get PLXS timestamps.

MCWDT is used to time simulations and LED blinking.

The design demonstrates the core functionality of the BLE Component configured as a PLXS device (GATT Server). Additionally, this project implements the services required by the PLXP specifications: BMS, BAS, DIS, and CTS as per the PLXP/PLXS v1.0 specification.

After a 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. `PlxsCallBack()`, `BmsCallBack()`, and `BasCallBack()` are used to receive events specific to the service attribute operations.

The server advertises in two modes:

- Non-bonded: The server advertises in the fast advertisement mode for 30 seconds, and falls back to slow advertisement mode for 150 seconds after a timeout.
- Bonded: In this mode, the bonded devices’ White List is enabled for the first 10 seconds of the advertisement. If the device is not connected within the first 10 seconds, the White List is disabled and advertisement starts in the discoverable undirected mode for the next 20 seconds. After 20 seconds, the mode automatically switches to slow advertisement for 150 seconds. On an advertisement timeout, the system enters Hibernate mode. Press **SW2** on the PSoC 6 BLE Pioneer Kit to wake up the system and restart advertising.

PLXS supports the Spot-check Measurement and Continuous Measurement characteristics. The Spot-check Measurement characteristic supports the SpO2, PR, timestamp, and pulse amplitude index features. The Continuous Measurement characteristic supports the SpO2PR-normal and pulse amplitude index features. This example project also supports the measurement storage for the Spot-check Measurement characteristic, which requires the Record Access Control Point (RACP) characteristic. The RACP characteristic supports the following procedures:

- Report Stored Records
- Report Number Stored Records
- Abort Operation
- Delete Stored Records

The RACP storage stores up to 30 PLX measurement records and overwrites older records as the operation progresses.

The PLXS simulation procedure simulates SpO2, PR, PAI, and timestamps values for the PLX Spot-check Measurement characteristic and PLX Continuous Measurement characteristic.

Simulation of the PLX Continuous Measurement characteristic data starts when connection is established and the notifications of the Continuous Measurement characteristic are enabled. The simulation ranges are: SpO2 95...100 percent (step: 1 percent), PR 50...100 bpm (step: 10 bpm), and PAI 10...20 percent (step: 0.15 percent). All range values and steps are defined in the “Simulations Defines” section of the *plxs.h* file.

Simulation of the PLX Spot-check Measurement characteristic data starts when the **Start Spot-check measurement** command is selected via the debug terminal press mechanical button (**SW2**). The simulation period is defined in `PLXS_SIM_SCMT_MT_PERIOD_COUNT` after the Spot-check simulation procedure stops. If the connection is established, simulated data is sent to the client via indications of the PLX Spot-check Measurement characteristic. Otherwise, the data is stored in the RACP storage. The simulation ranges are the same as for the Continuous Measurement characteristic, timestamps are retrieved from the RTC.

The Bond Manager Service (BMS) supports the procedures of BMS, Delete Bond of the Requesting Device, Delete all Bonds, and Delete Bond of all except the requesting device, for LE transport without an authorization key.

The Device Information Service (DIS) supports all characteristics required by Section 4.1.3 of the PLXS specification.

The Battery Service (BAS) is used for software simulation of the battery level. The simulated battery level is continuously changed from 2 to 20 percent.

The Current Time Service (CTS) is used to configure date and time in the Real Time Clock (RTC).

The blinking green LED on the BLE Pioneer kit indicates the advertising state., The red LED is turned ON to indicate the BLE disconnection state and blue LED is turned ON to indicate the BLE connected state.

The application enables SC (mode 1 level 4 option) with a passkey-based authenticated MITM and automatically falls back to the legacy authenticated MITM mode if secure connection (SC) is not supported by a peer device or the selected BLE device family.

To save power, the device switches to Deep Sleep mode between BLE connection intervals.

Table 1. Commands List

Command	Description
y	Send the **Accept the displayed passkey** command
n	Send the **Reject the displayed passkey** command

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 and 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 the Operation section.

The 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 and Dual core) in the BLE Peripheral Driver Library (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 core options.

Important to remember:

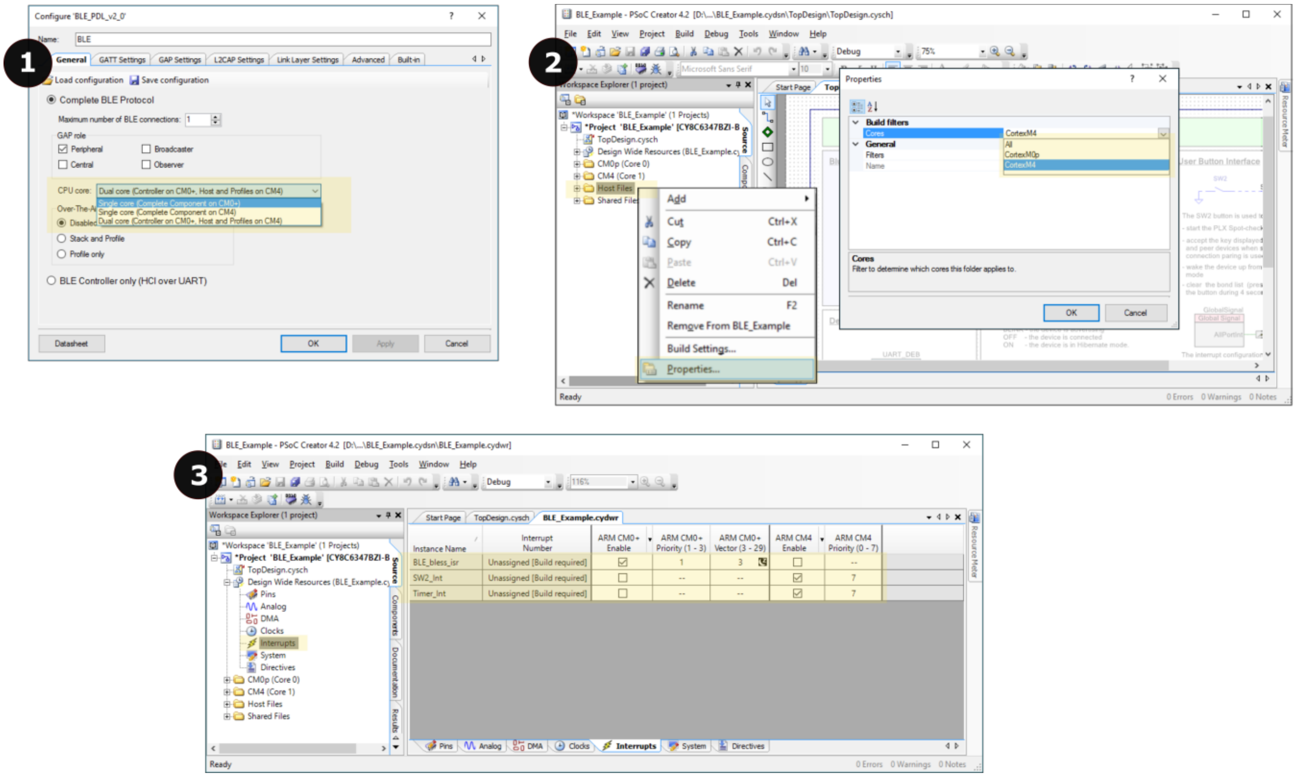
- All application host-files must be run on the host core.
- The BLE Subsystem (BLESS) interrupt must be assigned to the core where the controller runs.
- All additional interrupts (SW2, MCWDT, etc.) used in the example must be assigned to the host core.

Do the following to switch the CPU cores usage:

1. In the BLE Component Customizer **General** tab, select the appropriate CPU core option.
2. Change the core properties to CortexM4 or CortexC0p for the project folder Host Files based on the CPU core option selected 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 2. Steps for Switching the CPU Cores Usage



1. Configure 'BLE_PDL_v2_0'

Name: BLE

● Complete BLE Protocol

Maximum number of BLE connections: 1

GAP role: ☑ Peripheral ☐ Broadcaster ☐ Central ☐ Observer

CPU core: Dual core (Controller on CM0+, Host and Profiles on CM4)

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

Over-The-Air: ● Disabled ○ Stack and Profile ○ Profile only

○ BLE Controller only (HCI over UART)

2. BLE_Example - PSoC Creator 4.2 [D:\...\BLE_Example.cydsn\TopDesign\TopDesign.cysch]

Properties

Build filters

Cores: CortexM4 (All / CortexM0p / CortexM4)

General: Filters, Name

Cores
Filter to determine which cores this folder applies to.

3. BLE_Example - PSoC Creator 4.2 [D:\...\BLE_Example.cydsn\BLE_Example.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	Unassigned [Build required]	☑	1	3	☐	--
SW2_Int	Unassigned [Build required]	☐	--	--	☑	7
Timer_Int	Unassigned [Build required]	☐	--	--	☑	7

Hardware Setup

The code example was designed for the CY8CKIT-062 PSoC 6 BLE Pioneer Kit

Table 2 lists the required pin assignments and connections for the development board of 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]	

LED Behavior for V_{DDD} Voltage < 2.7 V

If the V_{DDD} voltage is set 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.

Components

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

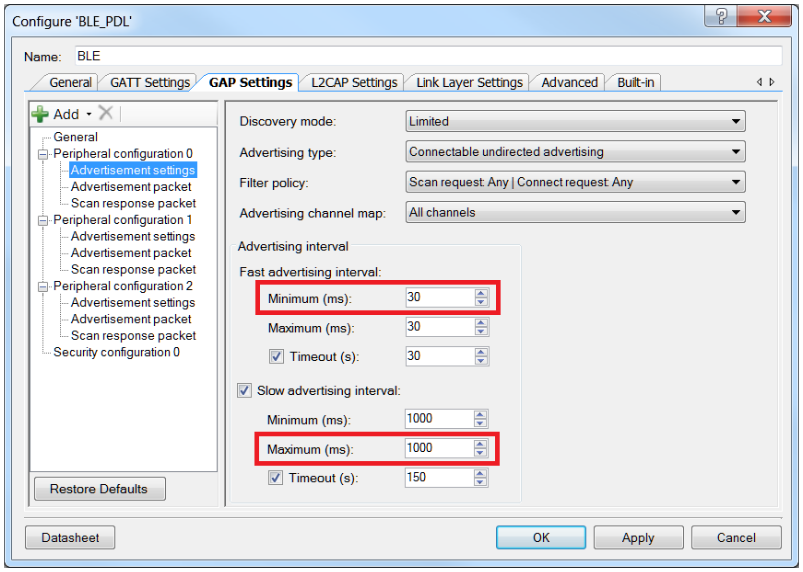
Table 3. PSoC Creator Components List

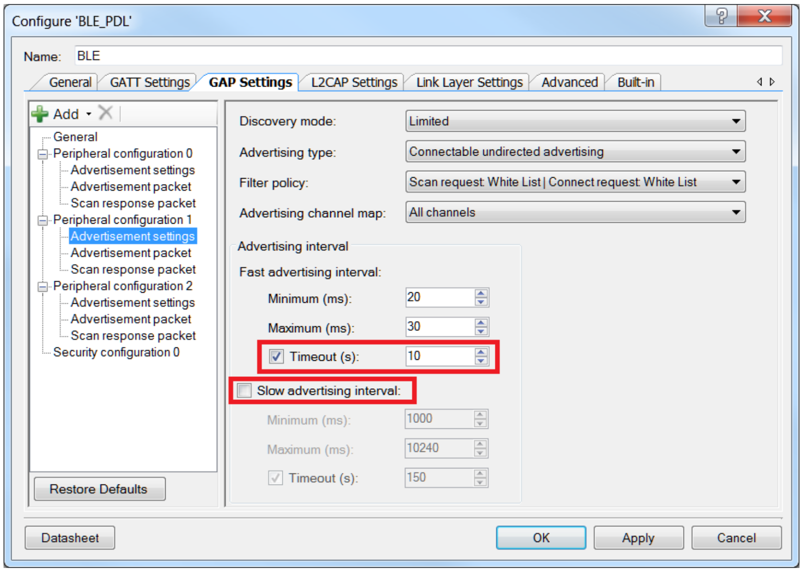
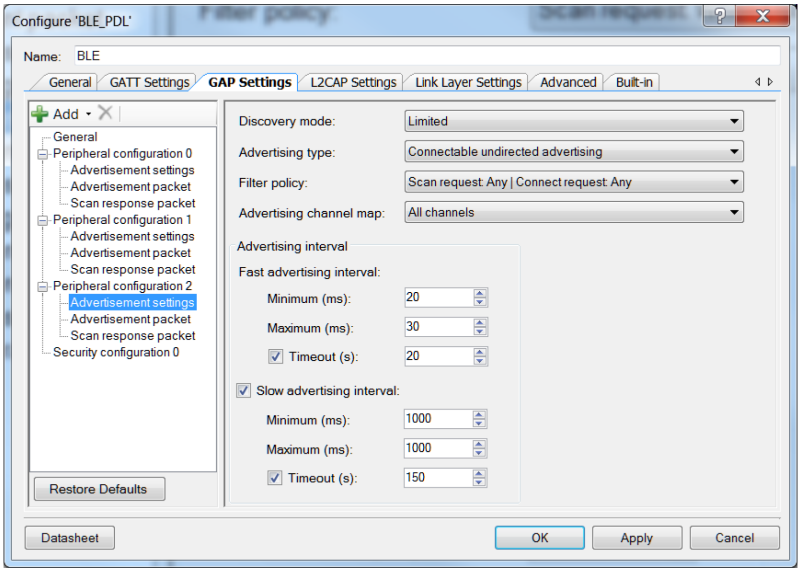
Component	Hardware Resources
UART_DEB	1 SCB
BLE	1 BLE, 1 Interrupt
SW2	1 pin
RTC	1 RTC
GlobalSignal	1 Interrupt
Wakeup_Interrupt	1 interrupt
Disconnect_LED Advertising_LED Simulation_LED	3 pins

Parameter Settings

The BLE Component is configured to support the GAP Peripheral role. The Component uses three advertisement configurations with the parameters listed in Table 4.

Table 4. GAP Settings > Advertisement Settings

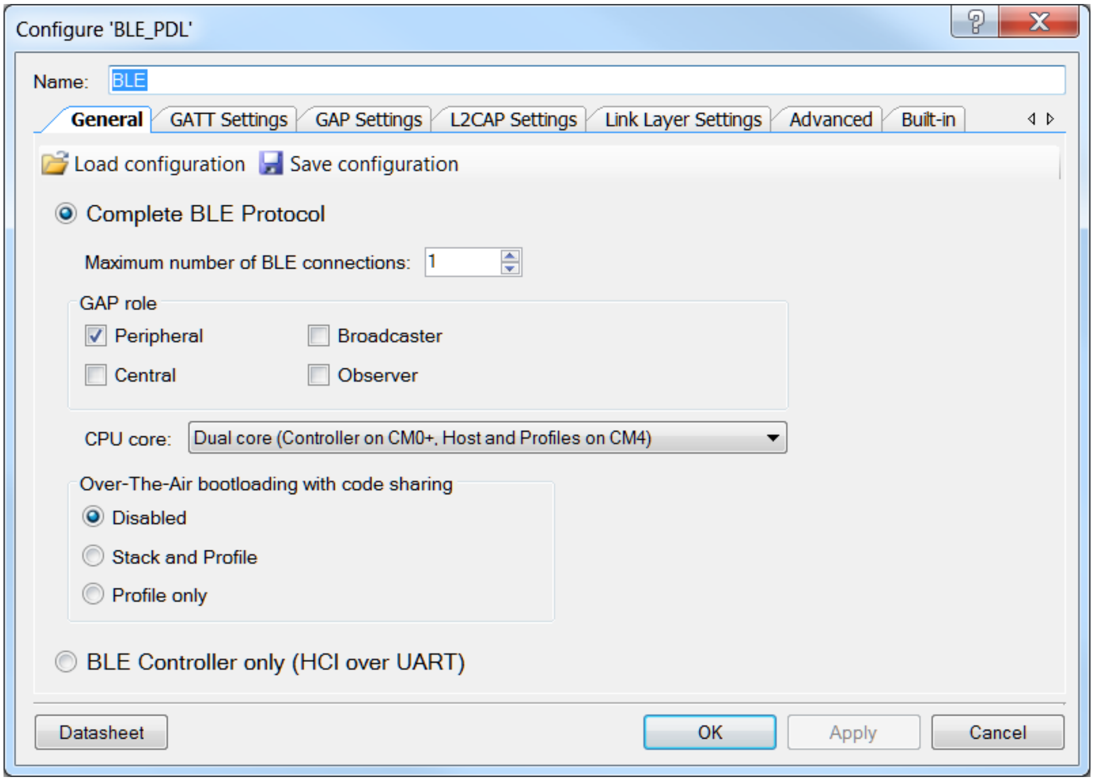
Advertise Mode	Advertise Configuration	Description
Non-bonded	CY_BLE_PERIPHERAL_ CONFIGURATION_0_INDEX	The design advertises in fast advertisement mode for 30 seconds and falls back to Slow Advertisement mode for 150 seconds after a timeout:  Configure 'BLE_PDL' General GATT Settings GAP Settings L2CAP Settings Link Layer Settings Advanced Built-in Add General Peripheral configuration 0 Advertisement settings Advertisement packet Scan response packet Peripheral configuration 1 Advertisement settings Advertisement packet Scan response packet Peripheral configuration 2 Advertisement settings Advertisement packet Scan response packet Security configuration 0 Restore Defaults Datasheet OK Apply Cancel Name: BLE Discovery mode: Limited Advertising type: Connectable undirected advertising Filter policy: Scan request: Any | Connect request: Any Advertising channel map: All channels Advertising interval Fast advertising interval: Minimum (ms): 30 Maximum (ms): 30 ☑ Timeout (s): 30 ☑ Slow advertising interval: Minimum (ms): 1000 Maximum (ms): 1000 ☑ Timeout (s): 150 On an advertisement timeout, the system enters Hibernate mode.

Bonded (Phase 1)	CY_BLE_PERIPHERAL_ CONFIGURATION_1_ INDEX	The bonded devices' White List (filter policy) is enabled for the first 10 seconds of the advertisement. On an advertisement timeout, the advertisement switches to Phase 2:  Configure 'BLE_PDL' Name: BLE Discovery mode: Limited Advertising type: Connectable undirected advertising Filter policy: Scan request: White List | Connect request: White List Advertising channel map: All channels Advertising interval Fast advertising interval: Minimum (ms): 20 Maximum (ms): 30 ☑ Timeout (s): 10 □ Slow advertising interval: Minimum (ms): 1000 Maximum (ms): 10240 ☑ Timeout (s): 150 General GATT Settings GAP Settings L2CAP Settings Link Layer Settings Advanced Built-in Add General Peripheral configuration 0 Advertisement settings Advertisement packet Scan response packet Peripheral configuration 1 Advertisement settings Advertisement packet Scan response packet Peripheral configuration 2 Advertisement settings Advertisement packet Scan response packet Security configuration 0 Restore Defaults Datasheet OK Apply Cancel
Bonded (Phase 2)	CY_BLE_PERIPHERAL_ CONFIGURATION_2_ INDEX	The White List is disabled and the advertisement starts in Discoverable Undirected mode for the next 20 seconds. After the 20 seconds, the mode automatically switches to slow advertisement mode for 150 seconds:  Configure 'BLE_PDL' Name: BLE Discovery mode: Limited Advertising type: Connectable undirected advertising Filter policy: Scan request: Any | Connect request: Any Advertising channel map: All channels Advertising interval Fast advertising interval: Minimum (ms): 20 Maximum (ms): 30 ☑ Timeout (s): 20 ☑ Slow advertising interval: Minimum (ms): 1000 Maximum (ms): 1000 ☑ Timeout (s): 150 General GATT Settings GAP Settings L2CAP Settings Link Layer Settings Advanced Built-in Add General Peripheral configuration 0 Advertisement settings Advertisement packet Scan response packet Peripheral configuration 1 Advertisement settings Advertisement packet Scan response packet Peripheral configuration 2 Advertisement settings Advertisement packet Scan response packet Security configuration 0 Restore Defaults Datasheet OK Apply Cancel On an advertisement timeout, the system enters Hibernate mode

The BLE Component is also configured to have the following:

- Public Device Address: 00A050-000021
- Device name: Pulse Oximeter
- Appearances: Pulse Oximeter: Fingertip
- Security Level: Authenticated LE Secure Connections pairing with encryption
- I/O capabilities: Display Yes/No
- Bonding requirements: Bonding

Figure 3. General Settings



Configure 'BLE_PDL'

Name: BLE

● Complete BLE Protocol

Maximum number of BLE connections: 1

GAP role

☑ Peripheral □ Broadcaster

□ Central □ Observer

CPU core: Dual core (Controller on CM0+, Host and Profiles on CM4)

Over-The-Air bootloading with code sharing

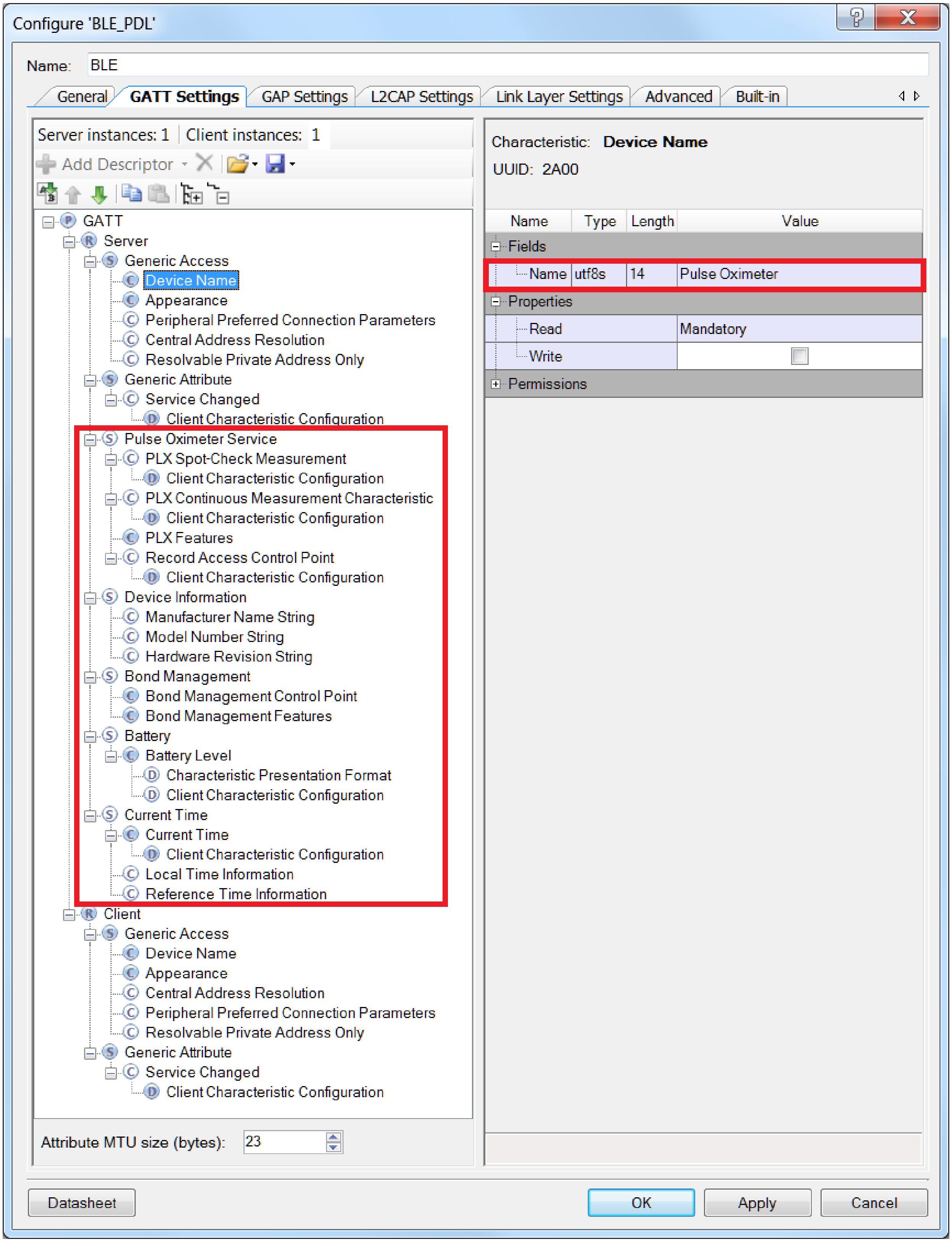
● Disabled

○ Stack and Profile

○ Profile only

○ BLE Controller only (HCI over UART)

Figure 4. GATT Settings



Name: BLE

Server instances: 1 | Client instances: 1

- GATT
  - Server
    - Generic Access
      - Device Name
      - Appearance
      - Peripheral Preferred Connection Parameters
      - Central Address Resolution
      - Resolvable Private Address Only
    - Generic Attribute
      - Service Changed
        - Client Characteristic Configuration
    - Pulse Oximeter Service
      - PLX Spot-Check Measurement
        - Client Characteristic Configuration
      - PLX Continuous Measurement Characteristic
        - Client Characteristic Configuration
      - PLX Features
      - Record Access Control Point
        - Client Characteristic Configuration
    - Device Information
      - Manufacturer Name String
      - Model Number String
      - Hardware Revision String
    - Bond Management
      - Bond Management Control Point
      - Bond Management Features
    - Battery
      - Battery Level
        - Characteristic Presentation Format
        - Client Characteristic Configuration
    - Current Time
      - Current Time
        - Client Characteristic Configuration
      - Local Time Information
      - Reference Time Information
  - Client
    - Generic Access
      - Device Name
      - Appearance
      - Central Address Resolution
      - Peripheral Preferred Connection Parameters
      - Resolvable Private Address Only
    - Generic Attribute
      - Service Changed
        - Client Characteristic Configuration

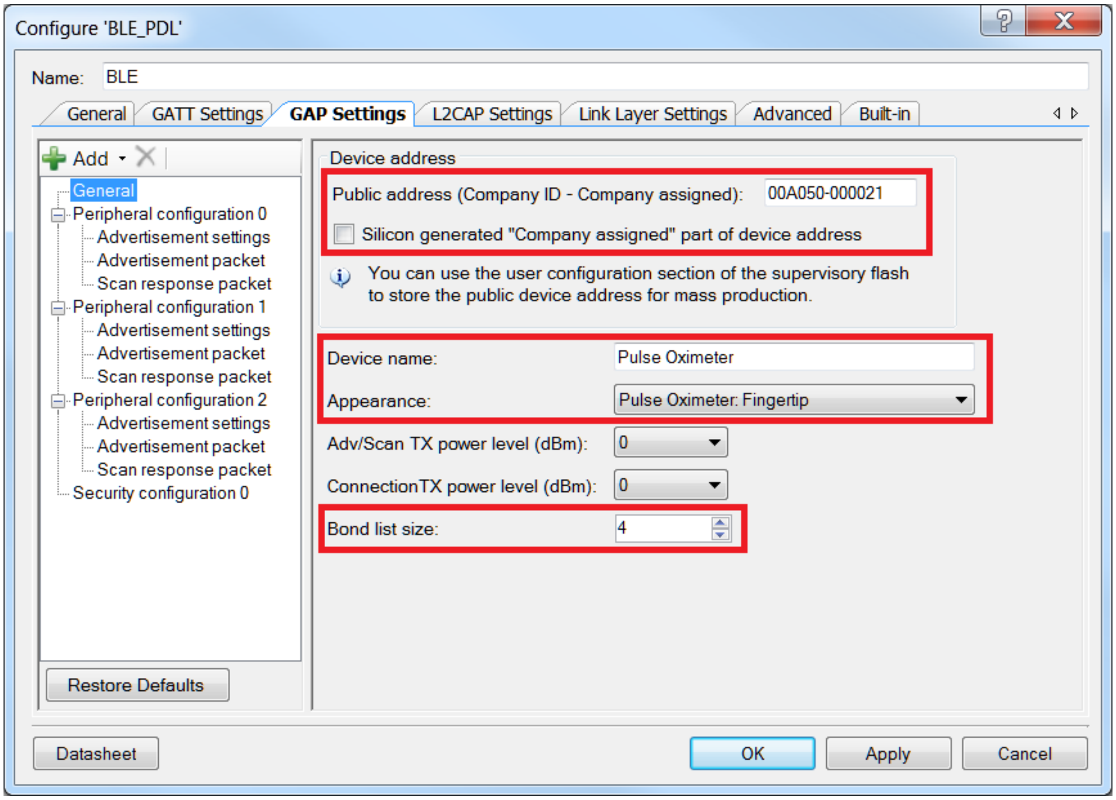
Attribute MTU size (bytes): 23

Characteristic: **Device Name**

UUID: 2A00

Name	Type	Length	Value
Fields			
Name	utf8s	14	Pulse Oximeter
Properties			
Read			Mandatory
Write			□
Permissions			

Figure 5. GAP Settings



Configure 'BLE_PDL'

Name: BLE

Device address

Public address (Company ID - Company assigned): 00A050-000021

□ Silicon generated "Company assigned" part of device address

You can use the user configuration section of the supervisory flash to store the public device address for mass production.

Device name: Pulse Oximeter

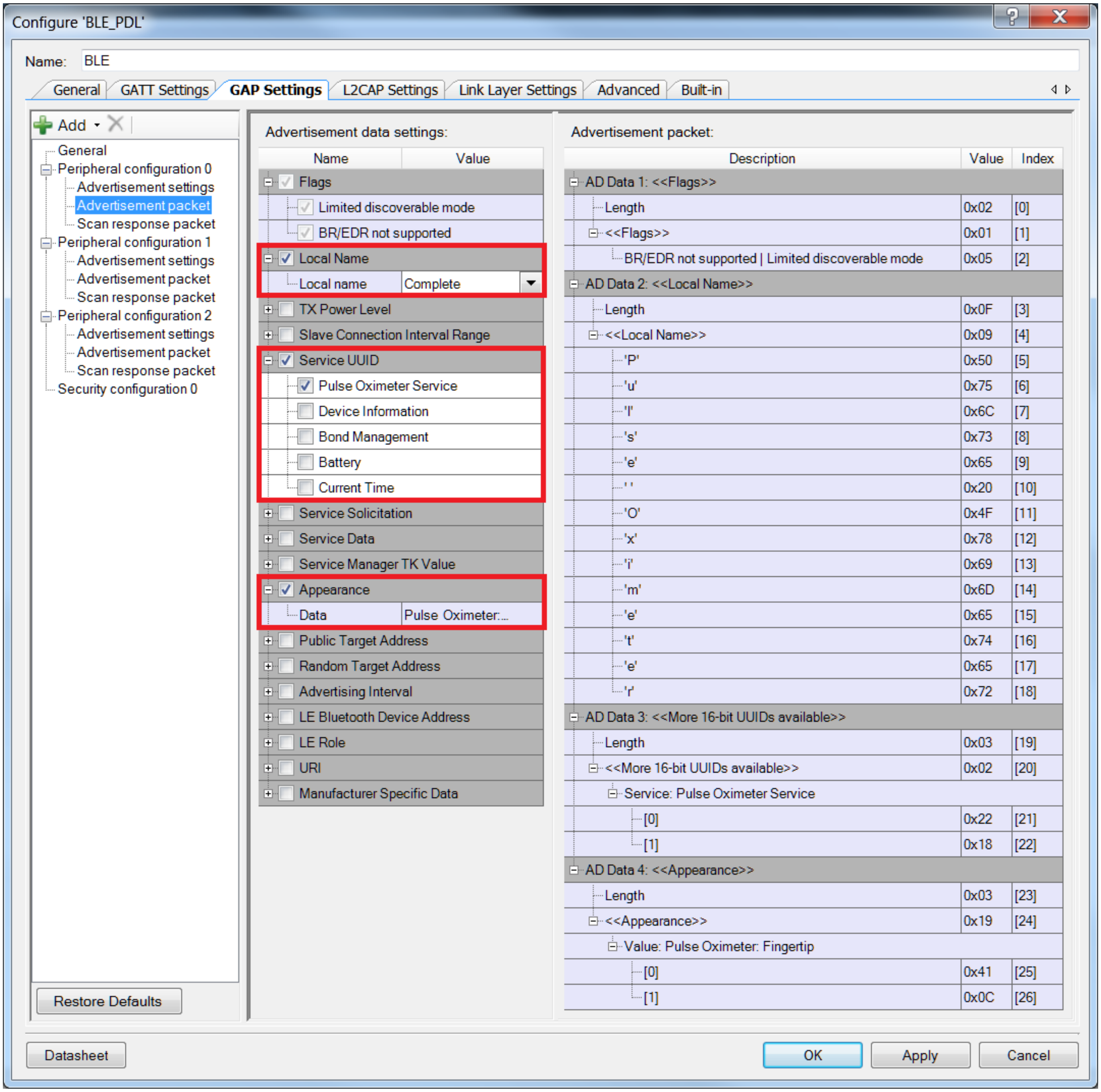
Appearance: Pulse Oximeter: Fingertip

Adv/Scan TX power level (dBm): 0

ConnectionTX power level (dBm): 0

Bond list size: 4

Figure 6. GAP Settings: Advertisement Packet



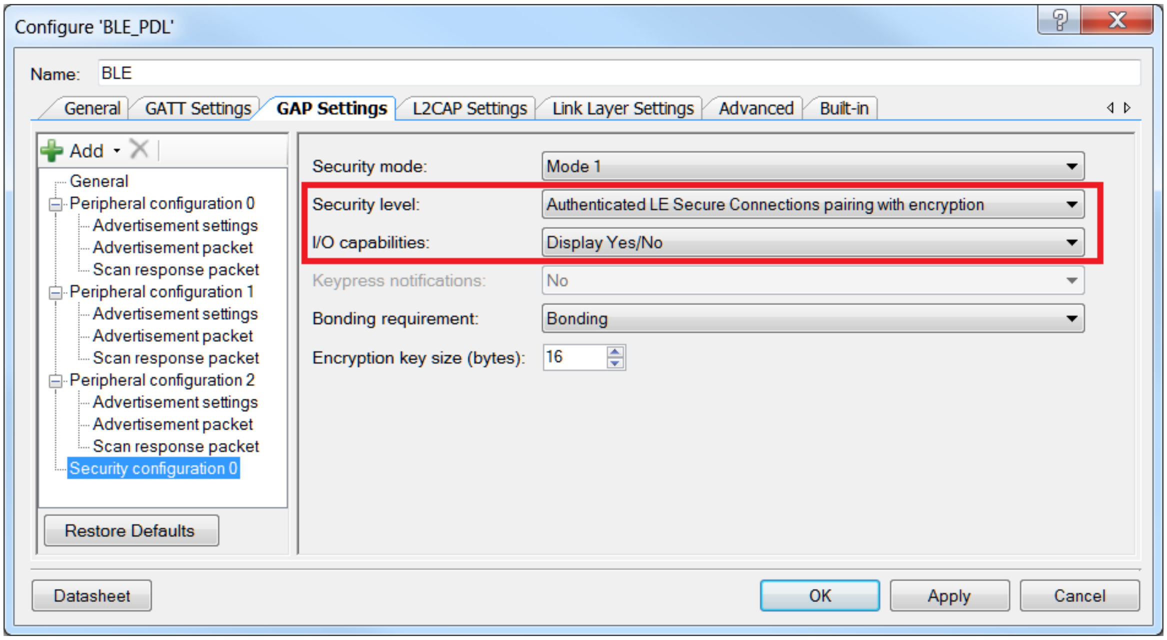
Advertisement data settings:

Name	Value
☑ Flags	
☑ Limited discoverable mode	
☑ BR/EDR not supported	
☑ Local Name	
Local name	Complete
☐ TX Power Level	
☐ Slave Connection Interval Range	
☑ Service UUID	
☑ Pulse Oximeter Service	
☐ Device Information	
☐ Bond Management	
☐ Battery	
☐ Current Time	
☐ Service Solicitation	
☐ Service Data	
☐ Service Manager TK Value	
☑ Appearance	
Data	Pulse Oximeter:...
☐ Public Target Address	
☐ Random Target Address	
☐ Advertising Interval	
☐ LE Bluetooth Device Address	
☐ LE Role	
☐ URI	
☐ Manufacturer Specific Data	

Advertisement packet:

Description	Value	Index
AD Data 1: <<Flags>>		
Length	0x02	[0]
<<Flags>>	0x01	[1]
BR/EDR not supported | Limited discoverable mode	0x05	[2]
AD Data 2: <<Local Name>>		
Length	0x0F	[3]
<<Local Name>>	0x09	[4]
'P'	0x50	[5]
'u'	0x75	[6]
'l'	0x6C	[7]
's'	0x73	[8]
'e'	0x65	[9]
' '	0x20	[10]
'O'	0x4F	[11]
'x'	0x78	[12]
'i'	0x69	[13]
'm'	0x6D	[14]
'e'	0x65	[15]
't'	0x74	[16]
'e'	0x65	[17]
'r'	0x72	[18]
AD Data 3: <<More 16-bit UUIDs available>>		
Length	0x03	[19]
<<More 16-bit UUIDs available>>	0x02	[20]
Service: Pulse Oximeter Service		
[0]	0x22	[21]
[1]	0x18	[22]
AD Data 4: <<Appearance>>		
Length	0x03	[23]
<<Appearance>>	0x19	[24]
Value: Pulse Oximeter: Fingertip		
[0]	0x41	[25]
[1]	0x0C	[26]

Figure 7. Security Settings



Name: BLE

Security mode: Mode 1

Security level: Authenticated LE Secure Connections pairing with encryption

I/O capabilities: Display Yes/No

Keypress notifications: No

Bonding requirement: Bonding

Encryption key size (bytes): 16

Operation

1. Prepare the setup:

- Connect the CySmart BLE Dongle to a USB port on the PC.
- Launch the CySmart Central Emulation Tool and select the connected dongle in the dialog window.
- Connect the BLE Pioneer board to a USB port on the PC, open the Device Manager, and note the COM port number for the KitProg USB-UART device in the ports (COM and LPT) branch of the tree.
- Build and program the BLE Pulse Oximeter project into the PSoC 6 BLE Pioneer Kit.
- Run a serial port communication program (Bray's Terminal, PuTTY, and so on.) and make a new connection to the noted COM port.

2. Connect to the PLXS device:

- Click **Start Scan** to discover available devices.
- Select the BLE Pulse Oximeter Project from the list of available devices and connect to it.
- Click **Yes** to a pairing request received from the peer device.
- Compare the displayed passkeys on both devices. Click **Yes** in CySmart and '**y**' on the terminal to confirm the Numeric comparison pairing procedure.
- Click **Yes** to add the device to the resolving list request from CySmart.

The output should appear as listed in Table 5:

Table 5. Terminal Output

Comment / Actions	Terminal log
	BLE Pulse Oximeter Sensor Project BLE Stack Version: 5.0.0.601
The RACP storage is empty	**INFO: plxsRacpOpr.storage is empty.** CY_BLE_EVT_STACK_ON, StartAdvertisement CY_BLE_EVT_SET_TX_PWR_COMPLETE CY_BLE_EVT_SET_TX_PWR_COMPLETE CY_BLE_EVT_SET_DEVICE_ADDR_COMPLETE CY_BLE_EVT_LE_SET_EVENT_MASK_COMPLETE CY_BLE_EVT_GAPP_ADVERTISEMENT_START_STOP, state: 2 CY_BLE_EVT_GAP_KEYS_GEN_COMPLETE CY_BLE_EVT_GAPP_ADVERTISEMENT_START_STOP, state: 1 CY_BLE_EVT_GAPP_ADVERTISEMENT_START_STOP, state: 2 CY_BLE_EVT_GATT_CONNECT_IND: 3, 7 CY_BLE_EVT_GAP_DEVICE_CONNECTED: connIntv = 8 ms CY_BLE_EVT_GATTS_XCNHG_MTU_REQ 3, 7, final mtu= 23 CY_BLE_EVT_GATTS_READ_CHAR_VAL_ACCESS_REQ: handle: 3 CY_BLE_EVT_L2CAP_CONN_PARAM_UPDATE_RSP, result = 0 CY_BLE_EVT_GAPC_CONNECTION_UPDATE_COMPLETE: 0, 2e(46.00 ms), a, ff CY_BLE_EVT_GAP_AUTH_REQ: bdHandle=7, security=3, bonding=1, ekeySize=10, err=0 CY_BLE_EVT_GAP_SMP_NEGOTIATED_AUTH_INFO: security:3, bonding:1, ekeySize:10, authErr 0 Compare this passkey with the passkey displayed in your peer device and press 'y' or 'n': 181988
Press '**y**' to accept the passkey	**y** Accept the displayed passkey CY_BLE_EVT_GAP_ENCRYPT_CHANGE: 0 CY_BLE_EVT_GAP_KEYINFO_EXCHNGE_CMPLT CY_BLE_EVT_GAP_AUTH_COMPLETE: security: 0x3, bonding: 0x1, ekeySize: 0x10, authErr 0x0 CY_BLE_EVT_PENDING_FLASH_WRITE

Start simulating data after connection	CY_BLE_EVT_ADD_DEVICE_TO_WHITE_LIST_COMPLETE CY_BLE_EVT_PENDING_FLASH_WRITE Simulated data: [Date: 01-01-0000 Time: 02:09:32] spO2: 99.00 PR: 90.00 PI: 10.60 Simulated data: [Date: 01-01-0000 Time: 02:09:35] spO2: 95.00 PR: 50.00 PI: 10.75 Simulated data: [Date: 01-01-0000 Time: 02:09:38] spO2: 96.00 PR: 60.00 PI: 10.90 Simulated data: [Date: 01-01-0000 Time: 02:09:41] spO2: 97.00 PR: 70.00 PI: 11.05 Simulated data: [Date: 01-01-0000 Time: 02:09:44] spO2: 98.00 PR: 80.00 PI: 11.20 Simulated data: [Date: 01-01-0000 Time: 02:09:47] spO2: 99.00 PR: 90.00 PI: 11.35 Simulated data: [Date: 01-01-0000 Time: 02:09:50] spO2: 95.00 PR: 50.00 PI: 11.50 Simulated data: [Date: 01-01-0000 Time: 02:09:53] spO2: 96.00 PR: 60.00 PI: 11.65 Simulated data: [Date: 01-01-0000 Time: 02:09:56] spO2: 97.00 PR: 70.00 PI: 11.80 Simulated data: [Date: 01-01-0000 Time: 02:09:59] spO2: 98.00 PR: 80.00 PI: 11.95

3. Read the PLX Continuous Measurement characteristic notifications.

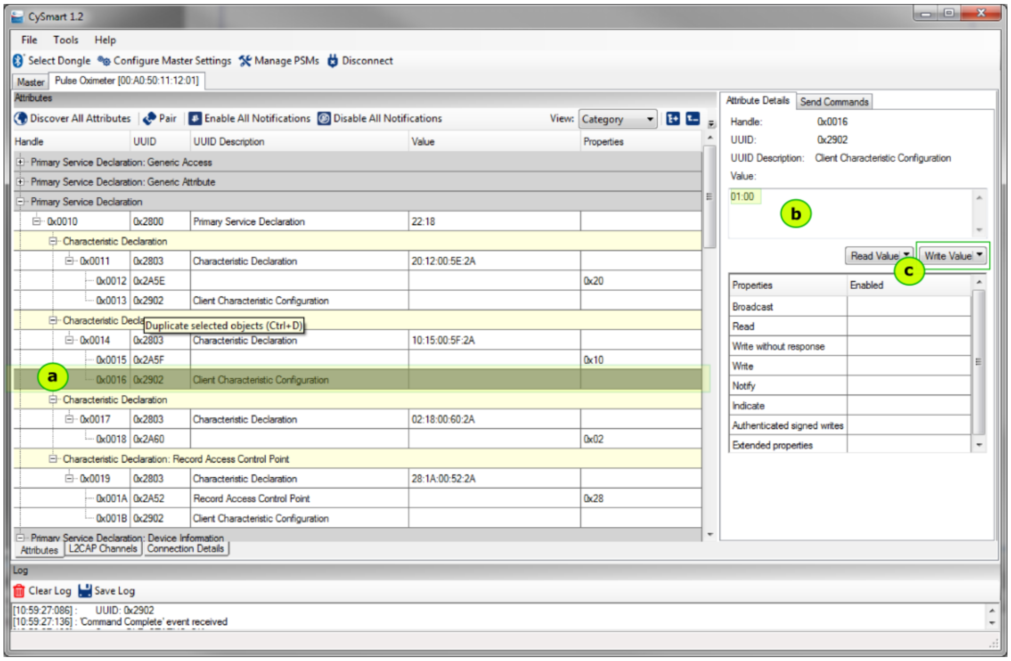
- In the CySmart Tool, enter **01:00** in the Client Characteristic Configuration (handle: 0x0016) of PLX Continuous Measurement characteristic to enable the notifications.

  a. Select the Client Characteristic Configuration (handle: 0x0016) characteristic.

  b. Enter **01:00** in the **Value** field of the **Attribute Details** tab.

  c. Click **Write Value** to send the command.

Figure 8. CySmart Windows Application: Writing Client Characteristic Configuration Characteristic



Handle	UUID	UUID Description	Value	Properties
Primary Service Declaration: Generic Access				
Primary Service Declaration: Generic Attribute				
Primary Service Declaration				
0x0010	0x2800	Primary Service Declaration	22:18	
Characteristic Declaration				
0x0011	0x2803	Characteristic Declaration	20:12:00:5E:2A	
0x0012	0x2A5E			0x20
0x0013	0x2902	Client Characteristic Configuration		
Characteristic Decla				
0x0014	0x2803	Characteristic Declaration	10:15:00:5F:2A	
0x0015	0x2A5F			0x10
0x0016	0x2902	Client Characteristic Configuration		
Characteristic Declaration				
0x0017	0x2803	Characteristic Declaration	02:18:00:60:2A	
0x0018	0x2A60			0x02
Characteristic Declaration: Record Access Control Point				
0x0019	0x2803	Characteristic Declaration	28:1A:00:52:2A	
0x001A	0x2A52	Record Access Control Point		0x28
0x001B	0x2902	Client Characteristic Configuration		
Primary Service Declaration: Device Information				

Handle: 0x0016

UUID: 0x2902

UUID Description: Client Characteristic Configuration

Value:

01:00

Properties	Enabled
Broadcast	
Read	
Write without response	
Write	
Notify	
Indicate	
Authenticated signed writes	
Extended properties	

```
[10:59:27:086] :      UUID: 0x2902
[10:59:27:136] : 'Command Complete' event received
```

The output appears as listed in Table 6:

Table 6. Terminal Output

Comments / Actions	Terminal log

Enable the notifications for the PLX Continuous Measurement Characteristic	`CY_BLE_EVT_PLXSS_NOTIFICATION_ENABLED:`
	`CY_BLE_EVT_GATTS_READ_CHAR_VAL_ACCESS_REQ: handle: 16`

Send the PLX Continuous Measurement notifications	`Simulated data: [ Date: 01-01-0000  Time: 02:10:05 ] spO2: 95.00 PR: 50.00 PI: 12.25`
	`INFO: the PLX Continuous Measurement characteristic was notified successfully`
	`Simulated data: [ Date: 01-01-0000  Time: 02:10:08 ] spO2: 96.00 PR: 60.00 PI: 12.40`
	`INFO: the PLX Continuous Measurement characteristic was notified successfully`
	`Simulated data: [ Date: 01-01-0000  Time: 02:10:11 ] spO2: 97.00 PR: 70.00 PI: 12.55`
	`INFO: the PLX Continuous Measurement characteristic was notified successfully`
	`Simulated data: [ Date: 01-01-0000  Time: 02:10:14 ] spO2: 98.00 PR: 80.00 PI: 12.70`
	`INFO: the PLX Continuous Measurement characteristic was notified successfully`
	`Simulated data: [ Date: 01-01-0000  Time: 02:10:17 ] spO2: 99.00 PR: 90.00 PI: 12.85`
	`INFO: the PLX Continuous Measurement characteristic was notified successfully`
	`…..`

4. Read the **PLX Spot-check Measurement characteristic** indications.

- In the CySmart tool, enter “**02:00**” in the Client Characteristic Configuration (handle: 0x0013) of the PLX Spot-check Measurement characteristic to enable the indication.
- Press **SW2** on the PSoC 6 BLE Pioneer Kit to start the PLX Spot-check Measurement simulation. In the debug terminal, observe that the simulated data indicates to the server:

```
Simulated data: [ Date: 07-02-0016  Time: 06:15:13 ] spO2: 96.00 PR: 60.00 PI: 13.90
INFO: the PLX Spot-check Measurement characteristic was indicated successfully
```

- Initiate disconnection on Client side (CySmart) to check if RACP storing.

- In the debug terminal, observe that the simulated data is added to the RACP storage:

```
Simulated data: [ Date: 07-02-0016 Time: 06:14:33 ] spO2: 98.00 PR: 80.00 PI: 11.95
INFO: the PLX Spot-check Measurement record was added to the RACP storage
```

The output should appear as listed in Table 7:

Table 7. Terminal Output

Comments / Actions	Terminal log

Enable the indication for the Spot-check Measurement Characteristic	`CY_BLE_EVT_GATTS_READ_CHAR_VAL_ACCESS_REQ: handle: 13`
	`CY_BLE_EVT_PLXSS_INDICATION_ENABLED`

Press ‘**SW2**’ to start the Spot-check session	**`INFO: start the Spot-check procedure`**
	`Simulated data: [ Date: 07-02-16  Time: 21:49:02 ] spO2: 97.00 PR: 70.00 PI: 11.80`
	`INFO: the PLX Spot-check Measurement characteristic was indicated successfully`

Simulated data indicates to the Client	`Simulated data: [ Date: 07-02-16  Time: 21:49:05 ] spO2: 98.00 PR: 80.00 PI: 11.95`
	`INFO: the PLX Spot-check Measurement characteristic was indicated successfully`
	….
	`Simulated data: [ Date: 07-02-16  Time: 21:49:23 ] spO2: 99.00 PR: 90.00 PI: 12.85`
	`INFO: the PLX Spot-check Measurement characteristic was indicated successfully`

Initiate disconnection on Client side	`CY_BLE_EVT_GATT_DISCONNECT_IND: 3, 7`
	`CY_BLE_EVT_GAP_DEVICE_DISCONNECTED: bdHandle=7, reason=16, status=0`
	`CY_BLE_EVT_GAPP_ADVERTISEMENT_START_STOP, state: 2`

Simulated data was stored in the RACP storage after disconnection	`Simulated data: [ Date: 07-02-16  Time: 21:49:26 ] spO2: 95.00 PR: 50.00 PI: 13.00`
	`INFO: The PLX Spot-check Measurement record was added to RACP storage`
	`Simulated data: [ Date: 07-02-16  Time: 21:49:29 ] spO2: 96.00 PR: 60.00 PI: 13.15`
	`INFO: The PLX Spot-check Measurement record was added to RACP storage`
	`Simulated data: [ Date: 07-02-16  Time: 21:49:32 ] spO2: 97.00 PR: 70.00 PI: 13.30`
	`INFO: The PLX Spot-check Measurement record was added to RACP storage`
	`Simulated data: [ Date: 07-02-16  Time: 21:49:35 ] spO2: 98.00 PR: 80.00 PI: 13.45`
	`INFO: The PLX Spot-check Measurement record was added to RACP storage`

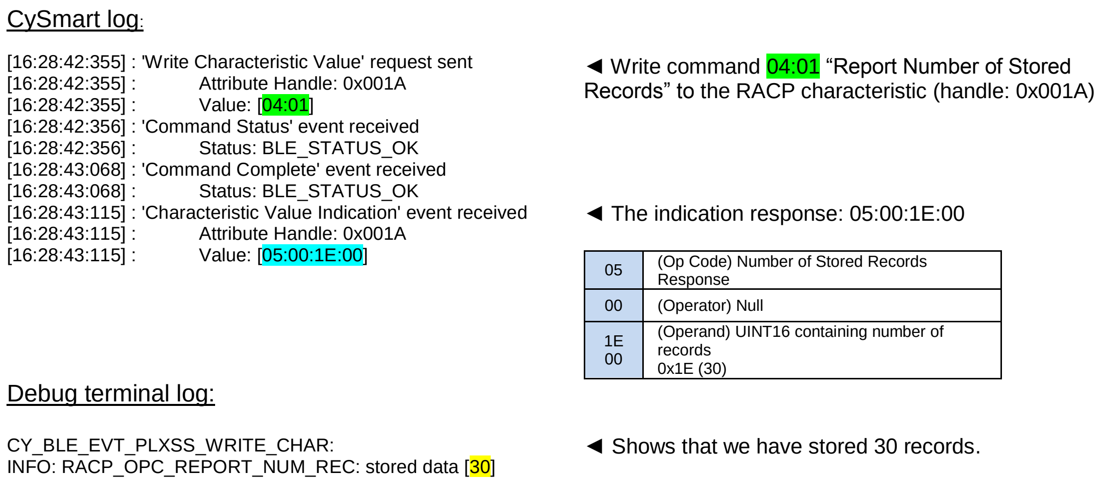
The simulation stops after a complete period of the Spot-check measurement (default 30 records)	`Simulated data: [ Date: 07-02-16  Time: 21:49:39 ] spO2: 99.00 PR: 90.00 PI: 13.60`
	`INFO: The PLX Spot-check Measurement record was added to RACP storage`
	`Simulated data: [ Date: 07-02-16  Time: 21:49:42 ] spO2: 95.00 PR: 50.00 PI: 13.75`

	INFO: The PLX Spot-check Measurement record was added to RACP storage … Simulated data: [Date: 07-02-16 Time: 21:50:45] spO2: 96.00 PR: 60.00 PI: 16.90 **INFO: The PLX Spot-check Measurement record was added to RACP storage** **INFO: finish the Spot-check procedure**

5. Perform RACP operations:
   For any RACP operation, enable the indications for the PLX Spot-check Measurement Characteristic and the Record Access Control Point Characteristic:

- In the CySmart tool, enter:
  - "**02:00**" in the Client Characteristic Configuration (handle: 0x0013) of the PLX Spot-check Measurement characteristic to enable an indication.
  - "**02:00**" in the Client Characteristic Configuration (handle: 0x001B) of the Record Access Control Point characteristic to enable an indication.
- Report Number Stored Records (op code: 04:01).
  - Write the **Report Number of Stored Records** command (value: 04:01) via the Record Access Control Point (RACP) characteristic (handle: 0x001A):
  - The output of the debug serial port communication program and the CySmart appears as shown in Figure 9.

Figure 9. CySmart Log and Terminal Output



CySmart log:

```
[16:28:42:355] : 'Write Characteristic Value' request sent
[16:28:42:355] :           Attribute Handle: 0x001A
[16:28:42:355] :           Value: [04:01]
[16:28:42:356] : 'Command Status' event received
[16:28:42:356] :           Status: BLE_STATUS_OK
[16:28:43:068] : 'Command Complete' event received
[16:28:43:068] :           Status: BLE_STATUS_OK
[16:28:43:115] : 'Characteristic Value Indication' event received
[16:28:43:115] :           Attribute Handle: 0x001A
[16:28:43:115] :           Value: [05:00:1E:00]
```

◄ Write command 04:01 "Report Number of Stored Records" to the RACP characteristic (handle: 0x001A)

◄ The indication response: 05:00:1E:00

05	(Op Code) Number of Stored Records Response
00	(Operator) Null
1E 00	(Operand) UINT16 containing number of records 0x1E (30)

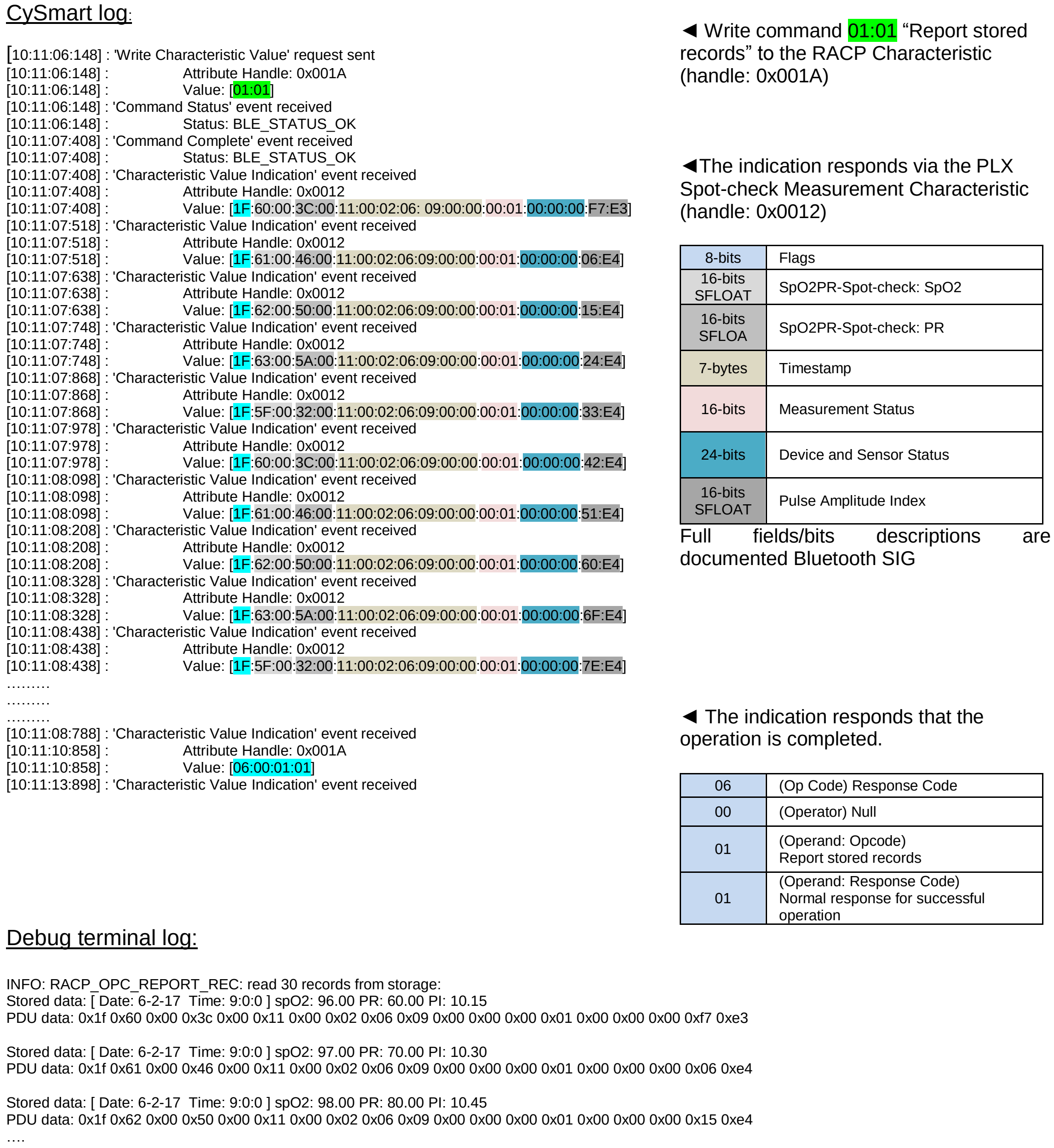
Debug terminal log:

```
CY_BLE_EVT_PLXSS_WRITE_CHAR:
INFO: RACP_OPC_REPORT_NUM_REC: stored data [30]
```

◄ Shows that we have stored 30 records.

- Report Stored Records (op code 01:01)
  - Write the **Report Stored Records** command (value 01:01) via the RACP characteristic (handle: 0x001A).`
  - The output of the debug serial port communication program and the CySmart should as shown in Figure 10.

Figure 10. CySmart Log and Terminal Output



CySmart log:

```
[10:11:06:148] : 'Write Characteristic Value' request sent
[10:11:06:148] :                Attribute Handle: 0x001A
[10:11:06:148] :                Value: [01:01]
[10:11:06:148] : 'Command Status' event received
[10:11:06:148] :                Status: BLE_STATUS_OK
[10:11:07:408] : 'Command Complete' event received
[10:11:07:408] :                Status: BLE_STATUS_OK
[10:11:07:408] : 'Characteristic Value Indication' event received
[10:11:07:408] :                Attribute Handle: 0x0012
[10:11:07:408] :                Value: [1F:60:00:3C:00:11:00:02:06: 09:00:00:00:01:00:00:00:F7:E3]
[10:11:07:518] : 'Characteristic Value Indication' event received
[10:11:07:518] :                Attribute Handle: 0x0012
[10:11:07:518] :                Value: [1F:61:00:46:00:11:00:02:06:09:00:00:00:01:00:00:00:06:E4]
[10:11:07:638] : 'Characteristic Value Indication' event received
[10:11:07:638] :                Attribute Handle: 0x0012
[10:11:07:638] :                Value: [1F:62:00:50:00:11:00:02:06:09:00:00:00:01:00:00:00:15:E4]
[10:11:07:748] : 'Characteristic Value Indication' event received
[10:11:07:748] :                Attribute Handle: 0x0012
[10:11:07:748] :                Value: [1F:63:00:5A:00:11:00:02:06:09:00:00:00:01:00:00:00:24:E4]
[10:11:07:868] : 'Characteristic Value Indication' event received
[10:11:07:868] :                Attribute Handle: 0x0012
[10:11:07:868] :                Value: [1F:5F:00:32:00:11:00:02:06:09:00:00:00:01:00:00:00:33:E4]
[10:11:07:978] : 'Characteristic Value Indication' event received
[10:11:07:978] :                Attribute Handle: 0x0012
[10:11:07:978] :                Value: [1F:60:00:3C:00:11:00:02:06:09:00:00:00:01:00:00:00:42:E4]
[10:11:08:098] : 'Characteristic Value Indication' event received
[10:11:08:098] :                Attribute Handle: 0x0012
[10:11:08:098] :                Value: [1F:61:00:46:00:11:00:02:06:09:00:00:00:01:00:00:00:51:E4]
[10:11:08:208] : 'Characteristic Value Indication' event received
[10:11:08:208] :                Attribute Handle: 0x0012
[10:11:08:208] :                Value: [1F:62:00:50:00:11:00:02:06:09:00:00:00:01:00:00:00:60:E4]
[10:11:08:328] : 'Characteristic Value Indication' event received
[10:11:08:328] :                Attribute Handle: 0x0012
[10:11:08:328] :                Value: [1F:63:00:5A:00:11:00:02:06:09:00:00:00:01:00:00:00:6F:E4]
[10:11:08:438] : 'Characteristic Value Indication' event received
[10:11:08:438] :                Attribute Handle: 0x0012
[10:11:08:438] :                Value: [1F:5F:00:32:00:11:00:02:06:09:00:00:00:01:00:00:00:7E:E4]
………
………
………
[10:11:08:788] : 'Characteristic Value Indication' event received
[10:11:10:858] :                Attribute Handle: 0x001A
[10:11:10:858] :                Value: [06:00:01:01]
[10:11:13:898] : 'Characteristic Value Indication' event received
```

◄ Write command 01:01 “Report stored records” to the RACP Characteristic (handle: 0x001A)

◄The indication responds via the PLX Spot-check Measurement Characteristic (handle: 0x0012)

8-bits	Flags
16-bits SFLOAT	SpO2PR-Spot-check: SpO2
16-bits SFLOA	SpO2PR-Spot-check: PR
7-bytes	Timestamp
16-bits	Measurement Status
24-bits	Device and Sensor Status
16-bits SFLOAT	Pulse Amplitude Index

Full fields/bits descriptions are documented Bluetooth SIG

◄ The indication responds that the operation is completed.

06	(Op Code) Response Code
00	(Operator) Null
01	(Operand: Opcode) Report stored records
01	(Operand: Response Code) Normal response for successful operation

Debug terminal log:

```
INFO: RACP_OPC_REPORT_REC: read 30 records from storage:
Stored data: [ Date: 6-2-17  Time: 9:0:0 ] spO2: 96.00 PR: 60.00 PI: 10.15
PDU data: 0x1f 0x60 0x00 0x3c 0x00 0x11 0x00 0x02 0x06 0x09 0x00 0x00 0x00 0x01 0x00 0x00 0x00 0xf7 0xe3

Stored data: [ Date: 6-2-17  Time: 9:0:0 ] spO2: 97.00 PR: 70.00 PI: 10.30
PDU data: 0x1f 0x61 0x00 0x46 0x00 0x11 0x00 0x02 0x06 0x09 0x00 0x00 0x00 0x01 0x00 0x00 0x00 0x06 0xe4

Stored data: [ Date: 6-2-17  Time: 9:0:0 ] spO2: 98.00 PR: 80.00 PI: 10.45
PDU data: 0x1f 0x62 0x00 0x50 0x00 0x11 0x00 0x02 0x06 0x09 0x00 0x00 0x00 0x01 0x00 0x00 0x00 0x15 0xe4
….
```

- Delete Stored Records (op code: 02:01)
  - Write the **Delete stored records** command (value: 02:01) via the Record Access Control Point (RACP) characteristic (handle: 0x001A).
  - The output of the debug serial port communication program and the CySmart appears as shown in Figure 11.

Figure 11. CySmart Log and Terminal Output

CySmart log:

```
[11:25:34:193] : 'Write Characteristic Value' request sent
[11:25:34:193] :        Attribute Handle: 0x001A
[11:25:34:193] :        Value: [02:01]
[11:25:34:193] : 'Command Status' event received
[11:25:34:193] :        Status: BLE_STATUS_OK
[11:25:34:843] : 'Command Complete' event received
[11:25:34:843] :        Status: BLE_STATUS_OK
[11:25:34:843] : 'Characteristic Value Indication' event received
[11:25:34:843] :        Attribute Handle: 0x001A
[11:25:34:843] :        Value: [06:00:02:01]
```

◄ Write command 02:01 **“Delete stored records”** to the RACP characteristic (handle: 0x001A).

◄ The indication response: 06:00:02:01

06	(Op Code) Response Code
00	(Operator) Null
02	(Operand: Opcode) Delete stored records
01	(Operand: Response Code) Normal response for successful operation

Debug terminal log:

```
CY_BLE_EVT_PLXSS_WRITE_CHAR:
INFO: RACP_OPC_DELETE_REC: remove all stored data
```

◄ Shows that all the records were removed.

- Abort Operation (op code: 03:00)
  - Write the **Abort operation** command (value: 03:00) via the Record Access Control Point (RACP) characteristic (handle: 0x001A).
  - The output of the debug serial port communication program and the CySmart appears as shown in Figure 12.

Figure 12. CySmart Log and Terminal Output

CySmart log:

```
[11:45:33:046] : 'Write Characteristic Value' request sent
[11:45:33:046] :        Attribute Handle: 0x001A
[11:45:33:046] :        Value: [03:00]
[11:45:33:046] : 'Command Status' event received
[11:45:33:046] :        Status: BLE_STATUS_OK
[11:45:33:766] : 'Command Complete' event received
[11:45:33:766] :        Status: BLE_STATUS_OK
[11:45:33:766] : 'Characteristic Value Indication' event received
[11:45:33:766] :        Attribute Handle: 0x001A
[11:45:33:766] :        Value: [06:00:03:01]
```

◄ Write command 03:00 **“Abort operation”** to the RACP characteristic (handle: 0x001A).

◄ The indication response: 06:00:03:01:

06	(Op Code) Response Code
00	(Operator) Null
03	(Operand: Opcode) Abort operation
01	(Operand: Response Code) Normal response for successful operation

Debug terminal log:

```
CY_BLE_EVT_PLXSS_WRITE_CHAR:
INFO: RACP_OPC_ABORT_OPN
```

◄ Shows that the **Abort** operation was triggered.

Related Documents

<table>
<tr><th colspan="3">Application Notes</th></tr>
<tr><td>AN210781</td><td>Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity</td><td>Describes the PSoC 6 MCU with BLE Connectivity, and how to build a basic code example.</td></tr>
<tr><td>AN215656</td><td>PSoC 6 MCU Dual-Core CPU System Design</td><td>Presents the theory and design considerations related to this code example.</td></tr>
<tr><th colspan="3">Software and Drivers</th></tr>
<tr><td colspan="2">CySmart – BLE Test and Debug Tool</td><td>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.</td></tr>
<tr><th colspan="3">PSoC Creator Component Datasheets</th></tr>
<tr><td colspan="2">Bluetooth Low Energy (BLE_PDL) Component</td><td>The Bluetooth Low Energy (BLE_PDL) Component provides a comprehensive GUI-based configuration window to facilitate designing applications requiring BLE connectivity.</td></tr>
<tr><th colspan="3">Device Documentation</th></tr>
<tr><td colspan="2">PSoC 6 MCU: PSoC 63 with BLE Datasheet Programmable System-on-Chip</td><td>PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual (TRM)</td></tr>
<tr><th colspan="3">Development Kit (DVK) Documentation</th></tr>
<tr><td colspan="3">CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit</td></tr>
</table>

Document History

Document Title: CE218044 - BLE Pulse Oximeter with PSoC 6 MCU with BLE Connectivity

Document Number: 002-18044

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5968185	NPAL	12/08/2017	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 MCU

Cypress Developer Community

Community | 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

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. 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.