PSoC 4 BLE Dynamic GATT Service Configuration

Objective

This example demonstrates how to dynamically enable or disable a GATT Service on the go.

Overview

In this example the BLE Component is configured as a GAP Peripheral and a GATT Server. It supports a custom Service called “RGB LED” through which a GAP Central device can control the color and brightness of the RGB LED on the BLE Pioneer Kit. This custom Service can be enabled or disabled dynamically by sending a command from the UART terminal to the BLE Pioneer Kit.

Requirements

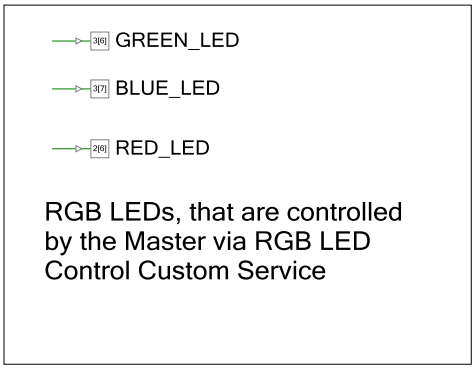
Design Tool: PSoC Creator 3.1 SP1, CySmart 1.0

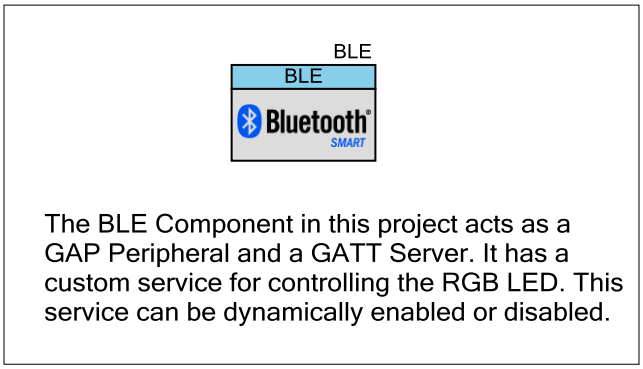
Programming Language: C (GCC 4.8.4 – included with PSoC Creator)

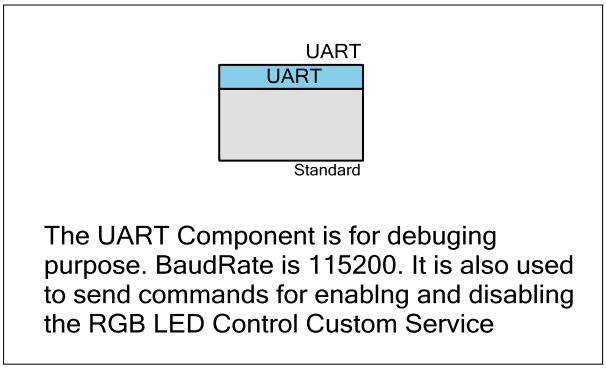
Associated Devices: All PSoC 4 BLE devices

Required Hardware: CY8CKIT-042-BLE Bluetooth® Low Energy (BLE) Pioneer Kit

PSoC Creator Schematic

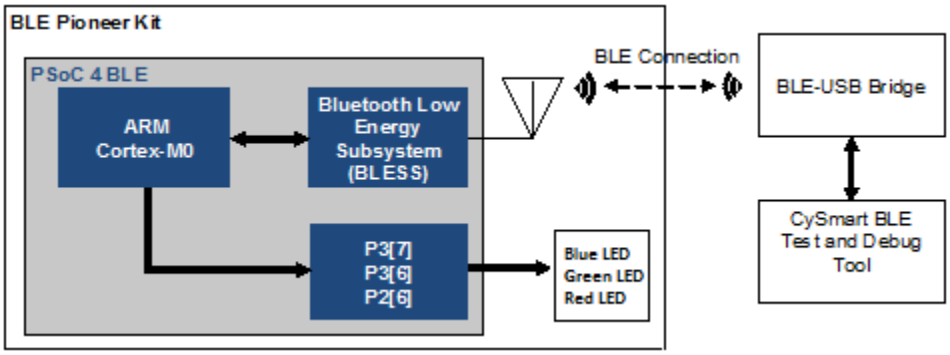






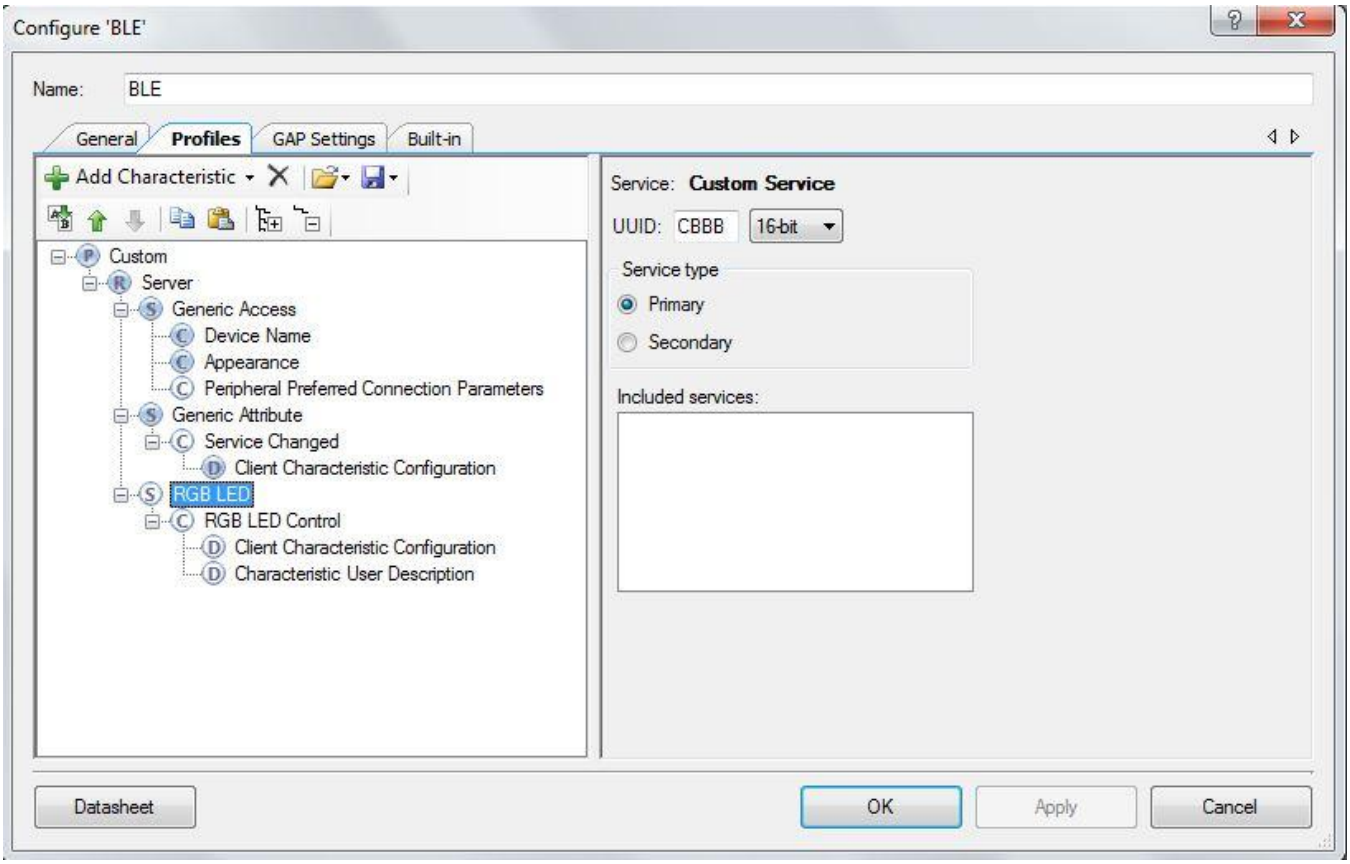
Hardware Setup

The BLE Pioneer Kit acts as a GAP Peripheral and the CySmart USB Dongle acts as the GAP Central.



Firmware:

The BLE Component is configured as a GAP Peripheral and a GATT Server. A custom Profile “RGB LED” is added



Initially, the BLE Component and the UART Component are turned on. The BLE Component starts Advertising as soon as the BLE Stack is ON. The while loop in the main function continuously scans for any inputs via the UART terminal. If 'D' is received, the **CyBle_GattsDisableAttribute** API is called for disabling the RGB LED custom Service. If 'E' is received, the **CyBle_GattsEnableAttribute** API is called for enabling the RGB LED custom Service. A "Service Changed" notification is sent to the GAP Central, if a connection is already established. This custom Service is enabled by default.

The custom Service enables the control of RGB LEDs from a GAP Central. When the GAP Central writes a value to the RGB LED custom Service Attribute, the **CYBLE_EVT_GATTS_WRITE_REQ** event is received by the GAP Peripheral. Upon the reception of this event, the data is extracted and the status of the LED is updated. Refer **Table 1** to see the list of values and the corresponding RGB LEDs state. Also, a response is sent to the GAP Central acknowledging the reception of the write request.

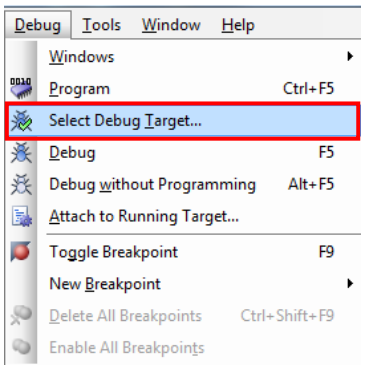
Build and Program

This section shows how to build the project and program the PSoC 4 BLE device on the BLE Pioneer Kit.

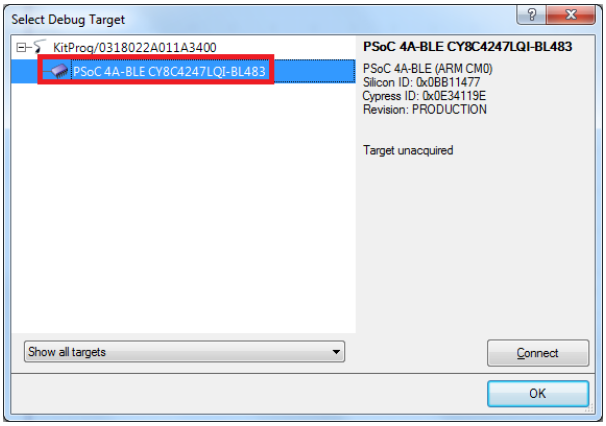
1. Open PSoC creator 3.1. Go to **File -> Open -> Project / Workspace**. Browse for the folder containing the project files and select **BLE_Dynamic_GATT_Service_Configuration.cyprj**.
2. Go to **Build -> Build BLE_ Dynamic_GATT_Service_Configuration.**
3. On a successful build, the total flash and SRAM usage is reported as shown below
4. Select **Debug** > **Select Debug Target**, as shown below.

```
Flash used: 72609 of 131072 bytes (55.4 %).
SRAM used: 12588 of 16384 bytes (76.8 %). Stack: 2048 bytes. Heap: 1024 bytes.
--------------- Build Succeeded: 03/14/2015 03:25:41 ---------------
```

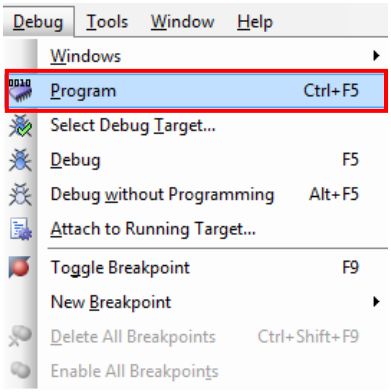
Selecting Debug Target



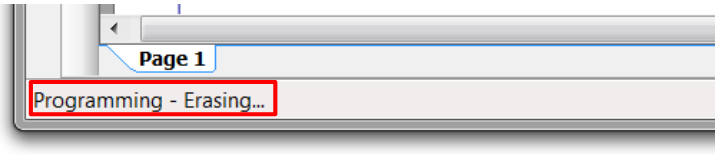
5. In the **Select Debug Target** dialog box, click **Port Acquire**, and then click **Connect** as shown below. Click **OK** to close the dialog box.



6. Select **Debug** > **Program** to program the device with the project, as shown below.



You can view the programming status on the PSoC Creator status bar (lower-left corner of the window), as shown below.

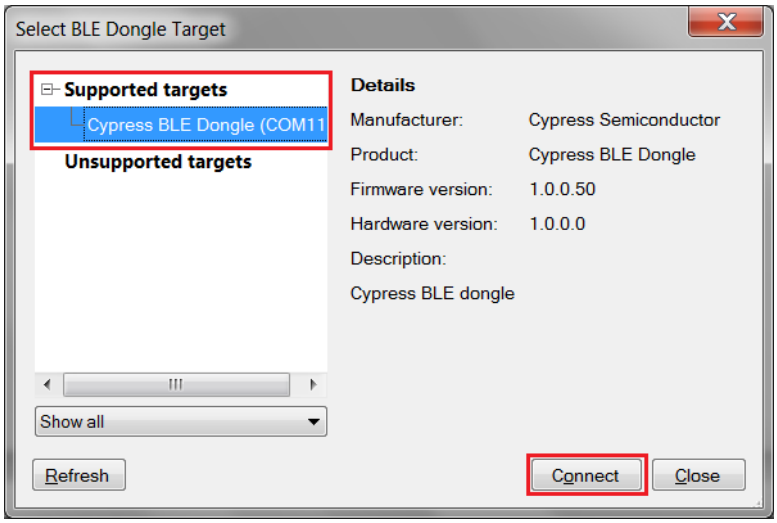


Operation and Testing

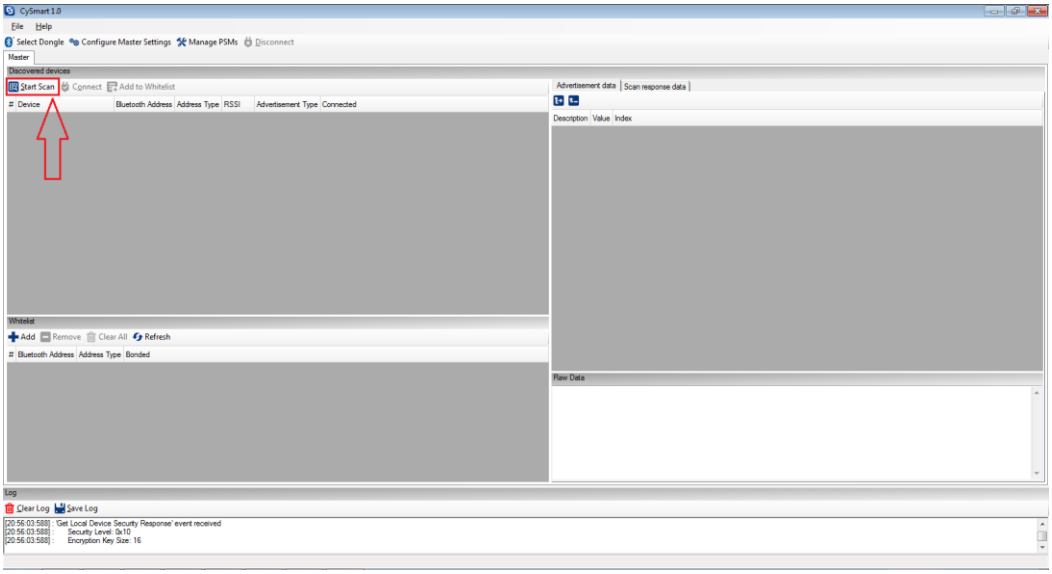
1) After programming, connected the kit to your PC, open a serial terminal (Putty or TeraTerm for example) corresponding to the BLE Pioneer Kit baseboard and configure the baud rate as 115200. Press the reset switch (SW1) on the kit. You can see the "**BLE Component ON"** and "**Starting to advertise"** messages in the terminal window.

2) On your PC, launch **CySmart 1.0**. It is located in the **All Programs -> Cypress -> CySmart** folder in the Windows start menu. The tool opens up and asks you to **Select BLE Dongle Target**. Select the **Cypress BLE Dongle (COMxx)** and click **Connect**, as shown in below.

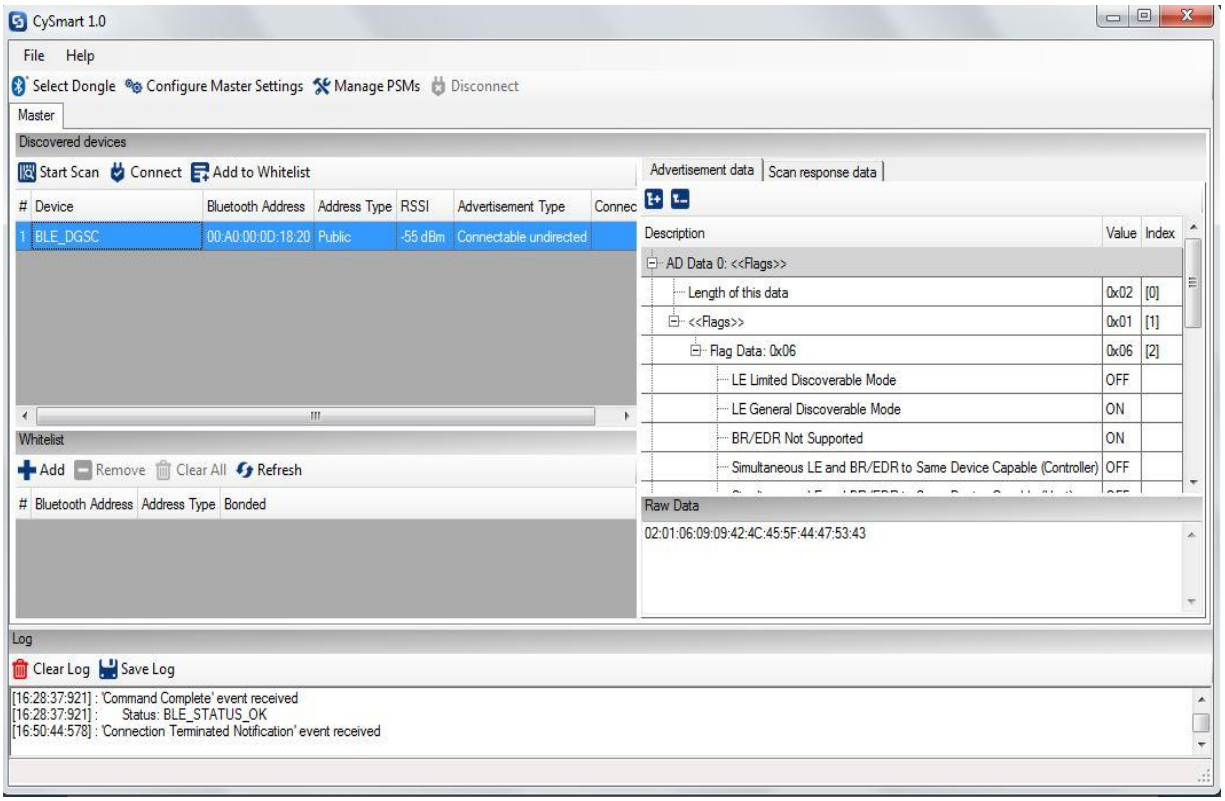
CySmart: Select BLE Dongle Target



3) When the BLE Dongle is connected, click on **Start Scan** to search for your BLE device as shown below.

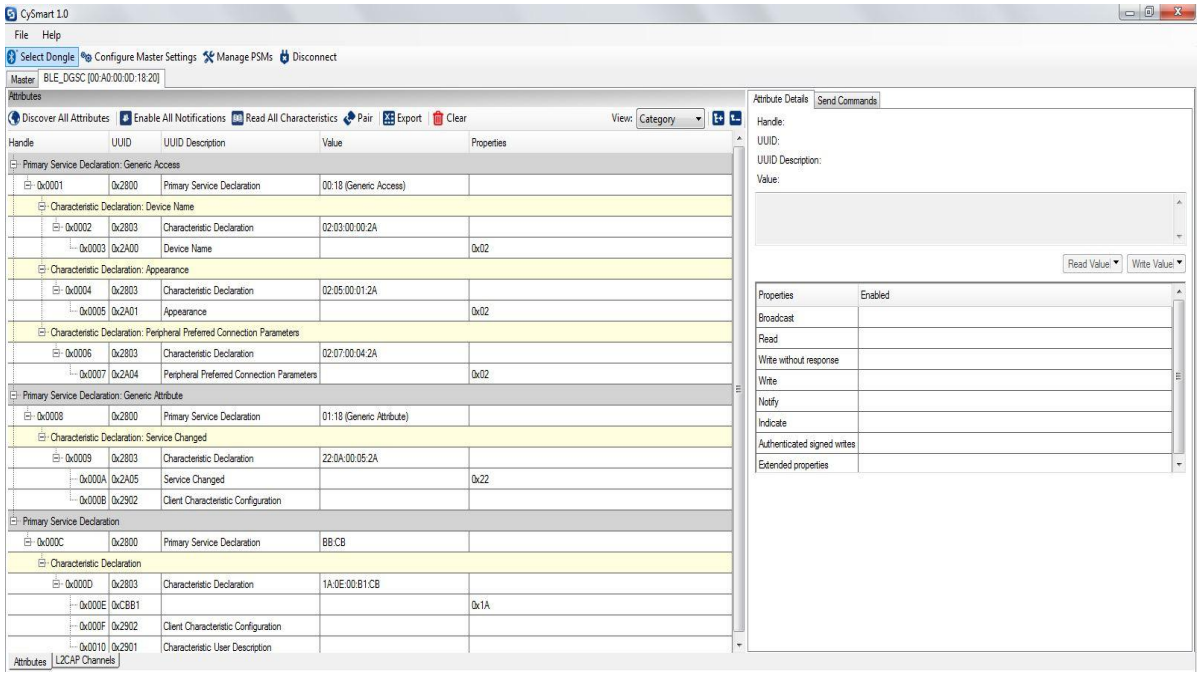


4) You can see the "**BLE_DGSC**" device detected as shown below. Click the **Connect** button.



5) After connection, click **Discover All Attributes** and "**Enable All Notifications"** buttons. The window will now appear as shown below. You can see that three (3) Services are enabled as shown in the figure below. The first two Services

are the default Generic Access Profile (GAP) and Generic Attribute Profile (GATT). The third one is the RGB LED custom Service included in this example. This shows that the third Service is currently enabled.

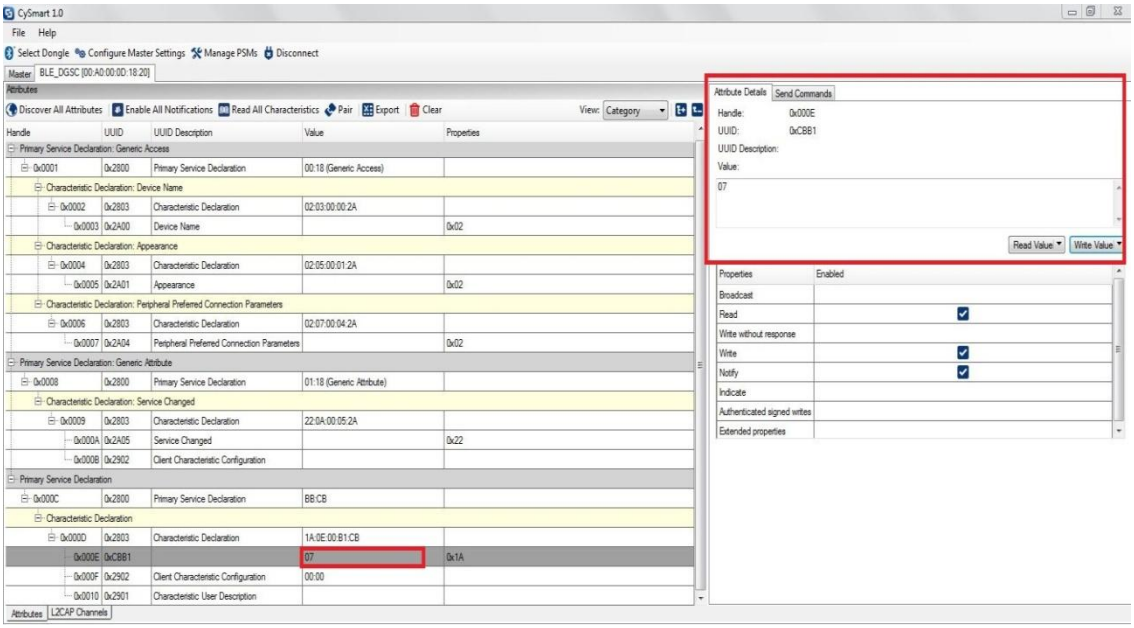


6) The status of the RGB LEDs can be changed by CySmart using this custom Service. Select the row corresponding to the handle 0x000E. On the right pane, in the **Attribute Details** Tab, enter a number from 1 to 7 and click **Write Value**. You will see the state of RGB LEDs change on the BLE Pioneer Kit baseboard as per the Table below:

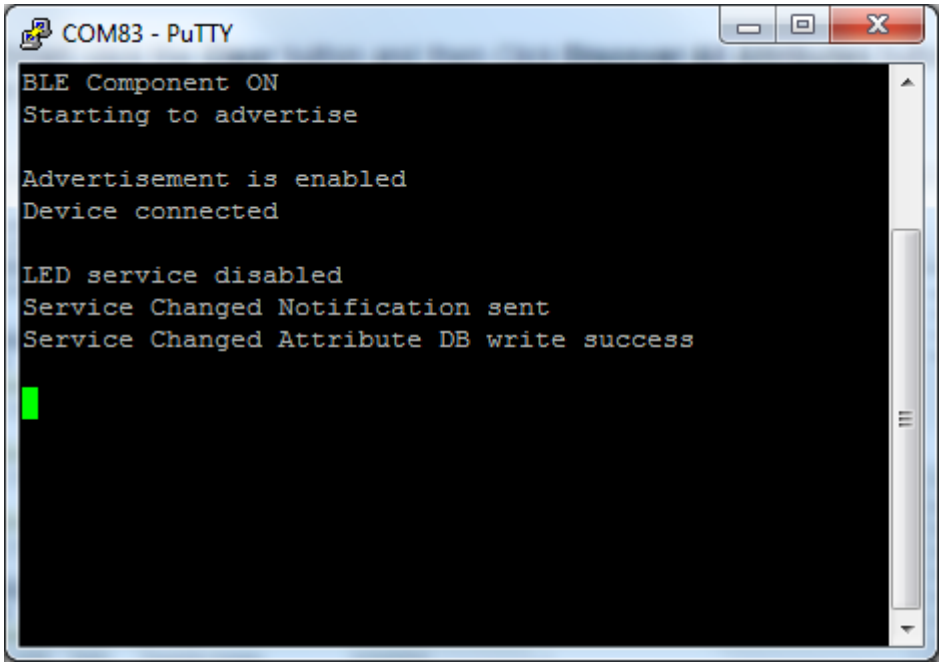
Table 1: RGB LED Custom Characteristic Value and the State of the LEDs

Value	RED Led	Green LED	BLUE Led	Resultant Color
0	OFF	OFF	OFF	-
1	ON	OFF	OFF	RED
2	OFF	ON	OFF	GREEN
3	ON	ON	OFF	YELLOW
4	OFF	OFF	ON	BLUE
5	ON	OFF	ON	MAGENTA
6	OFF	ON	ON	CYAN
7	ON	ON	ON	WHITE

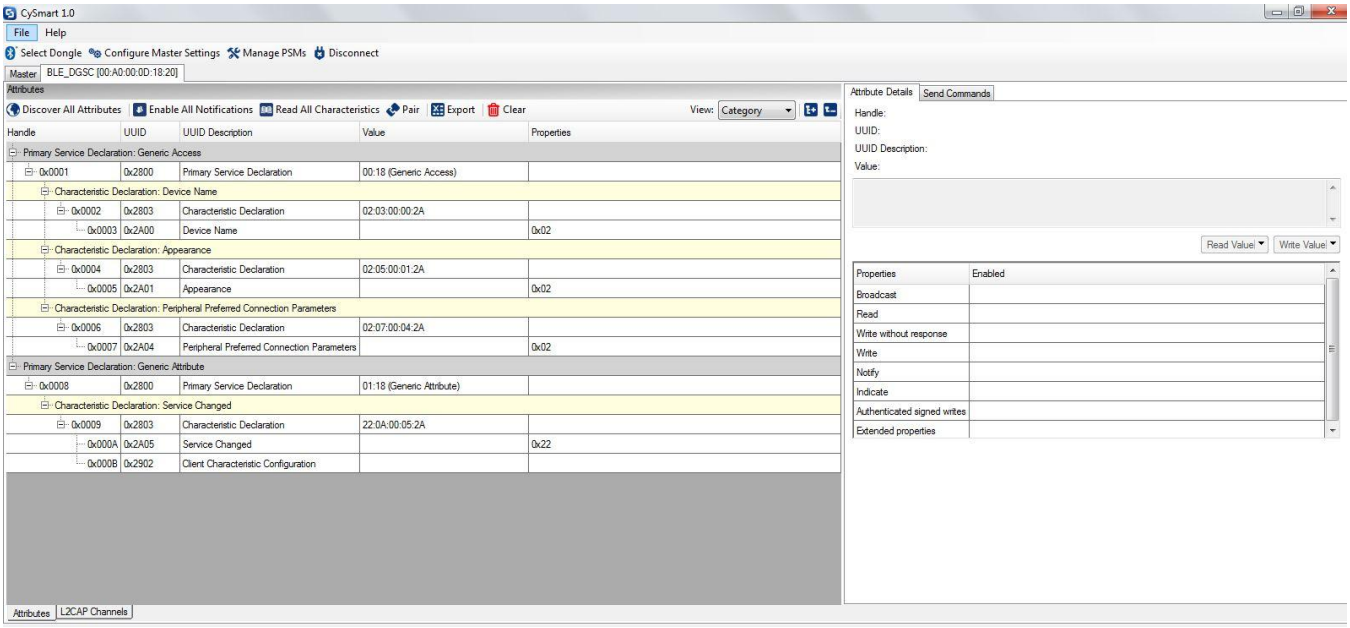
Also, when read, the new value will be updated on the CySmart as shown below:



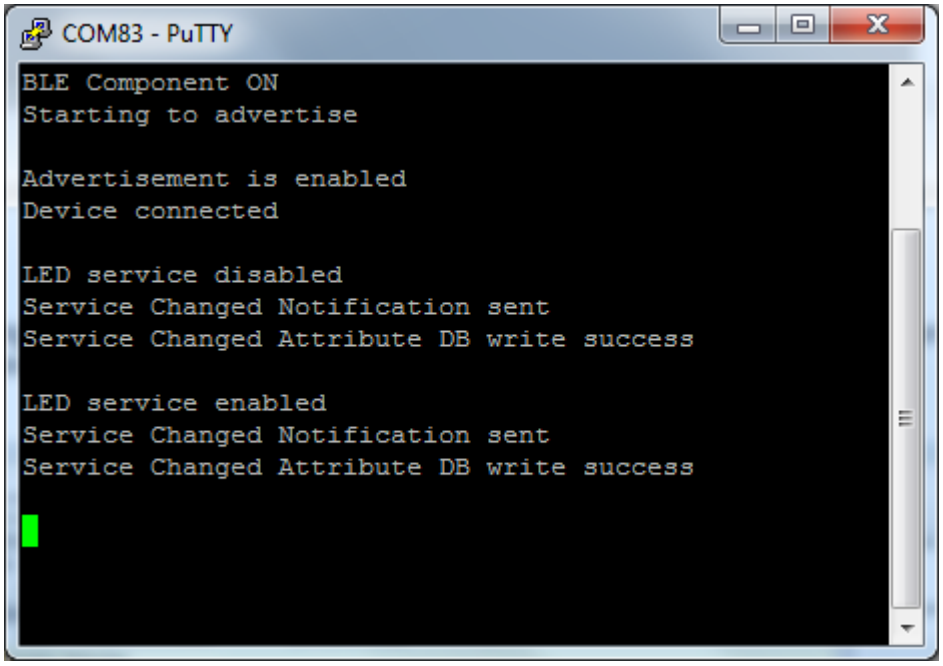
7) To disable the service, Enter 'D' from the UART terminal. The debug messages will be displayed as shown.



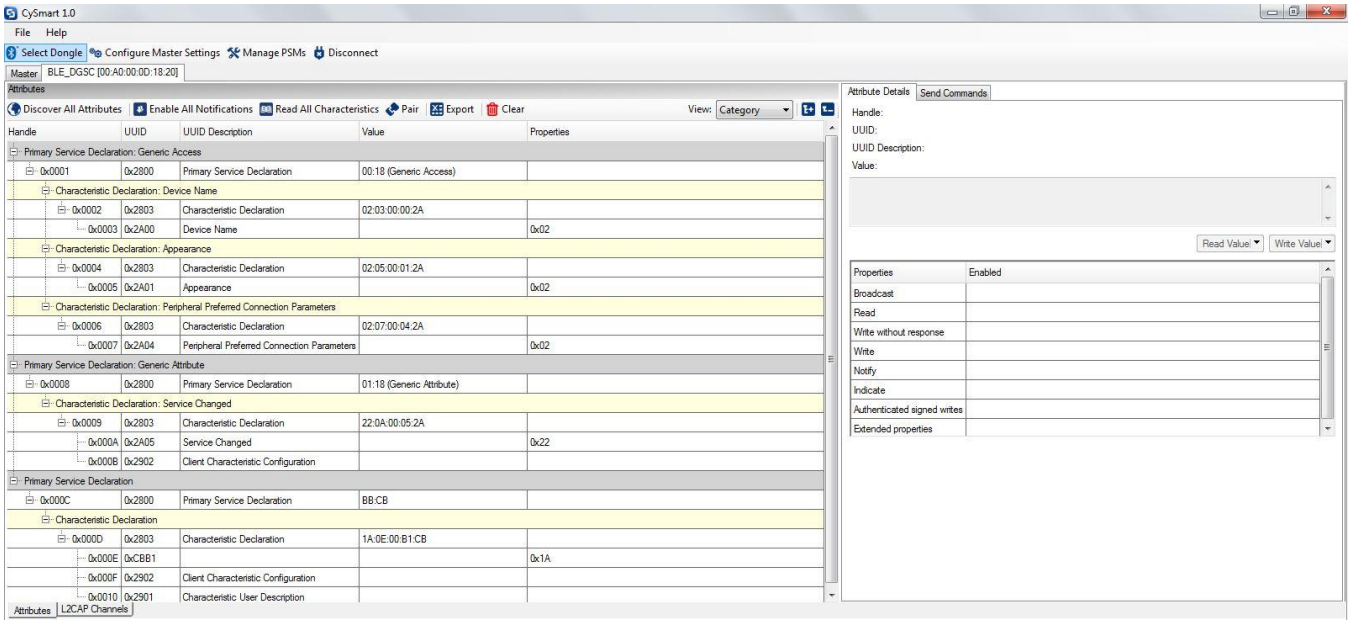
8) In CySmart, click the **Clear** button and then Click **Discover All Attributes** button. This time, the custom Service will not be shown.



9) To enable the service again, enter ‘E’ on the UART terminal. The debug messages will be displayed as shown below:



10) Clicking **Discover All Attributes** button on CySmart again will show the Service indicating that it has been re-enabled.



Related Documents

The table below lists all relevant application notes, code examples, knowledge base articles, device datasheets, and Component datasheets.

Related Documents

Document	Title	Comment
AN91267	Getting Started with PSoC® 4 BLE	A guide for beginners on PSoC 4 BLE
001-90479	Programmable System-on-Chip (PSoC®)	PSoC® 4: PSoC 4XX7_BLE Family Datasheet