PSoC 4 BLE - Bonding

Objective

This example demonstrates the Bonding functionality of the BLE Component in PSoC Creator IDE using the PSoC 4 BLE device.

Overview

Bonding is the process in which the keys and identity information exchanged during the pairing process for securing and encrypting the link are saved. After devices are bonded, they do not have to go through the pairing process again when reconnected. In this example, the BLE Component requires the peer device to enter a passkey for establishing a connection for the first time. Once the connection is established, this passkey and identity information is stored and device is said to be bonded. Once bonded, no passkey is required upon reconnection, until the bonding data is removed manually by the user. There can be a maximum of 4 bonded devices.

Requirements

Design Tool: PSoC Creator 3.1 CP1, CySmart 1.0

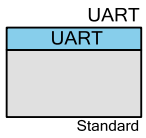
Programming Language: C (GCC 4.8.4 – included with PSoC Creator)

Associated Devices: All PSoC 4 BLE devices

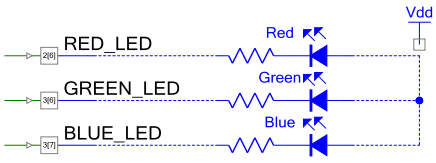
Required Hardware: CY8CKIT-042-BLE Bluetooth® Low Energy (BLE) Pioneer Kit

PSoC Creator Schematic

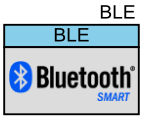
Figure 1. PSoC Creator Schematic



UART is used for transmitting the debug information.



Green LED is used to indicate that device is advertising.
Red LED is used to indicate that device is in disconnection state.
Blue LED is used to indicate that Authentication succeeded

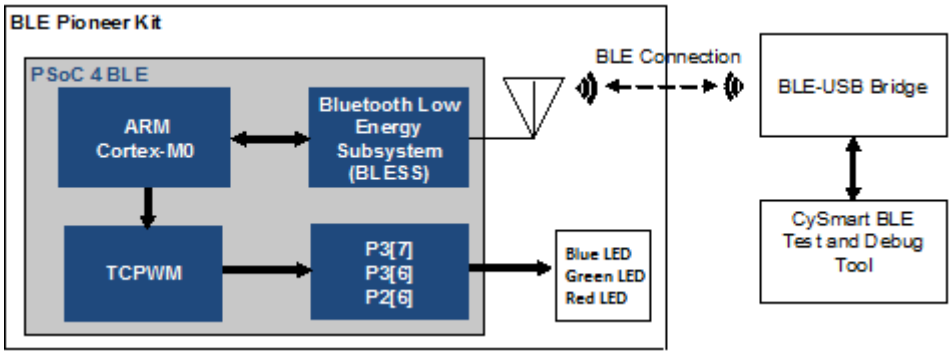


BLE component configured as a Peripheral and a Client requires the peer device to enter Passkey for Authentication.
It is capable of Bonding (Storig the pairing information for future use)

Hardware Setup

The BLE Component in a CY8CKIT-042-BLE Kit acts as a GAP Peripheral and the BLE-USB bridge acts as a GAP Central.

Figure 2. Hardware Setup



Firmware

In this example, the BLE Pioneer Kit, acting as the GAP Peripheral and the GATT Client, starts advertising as soon the BLE stack is turned on. When a connection is established with a peer device, it discovers the GATT Server and sends the authentication request to the peer device. As the authentication is performed by entering a passkey, the peer device sends request for displaying the passkey, following which the BLE Component displays the passkey in the UART Serial Terminal. When the correct passkey is entered from the peer device, the authentication succeeds and the pairing information is automatically stored in the flash, which would enable the devices to reconnect without going through the process again.

For removing the bonding information, the disconnection should be initiated by the peer device. After disconnection, the BLE Component automatically starts advertising. The while loop in the main function continuously keeps scanning for character received from the UART. If 'R' received, it stops the advertisement and removes the device from the bonding list.

Build and Program

This section shows how to build the project and program the PSoC 4 BLE device. If you are using a development kit with a built-in programmer (BLE Pioneer Kit, for example), connect the BLE Pioneer Baseboard to your computer using the USB Standard-A to Mini-B cable. For other kits, refer to the kit user guide.

If you are developing on your own hardware, you need a hardware debugger, for example, a Cypress CY8CKIT-002 MiniProg3.

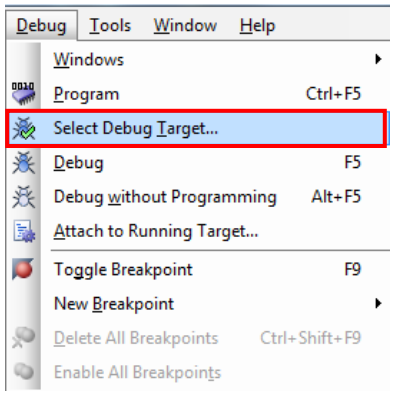
1. On PSoC Creator, select Build > Clean and Build **BLE_Bonding.**
2. On a successful build, the total flash and SRAM usage is reported as shown below.

Figure 3. Reporting Successful build

```
Flash used: 80575 of 131072 bytes (61.5 %).
SRAM used: 12876 of 16384 bytes (78.6 %). Stack: 1536 bytes. Heap: 1792 bytes.
---------------- Build Succeeded: 02/25/2015 14:14:37 ----------------
```

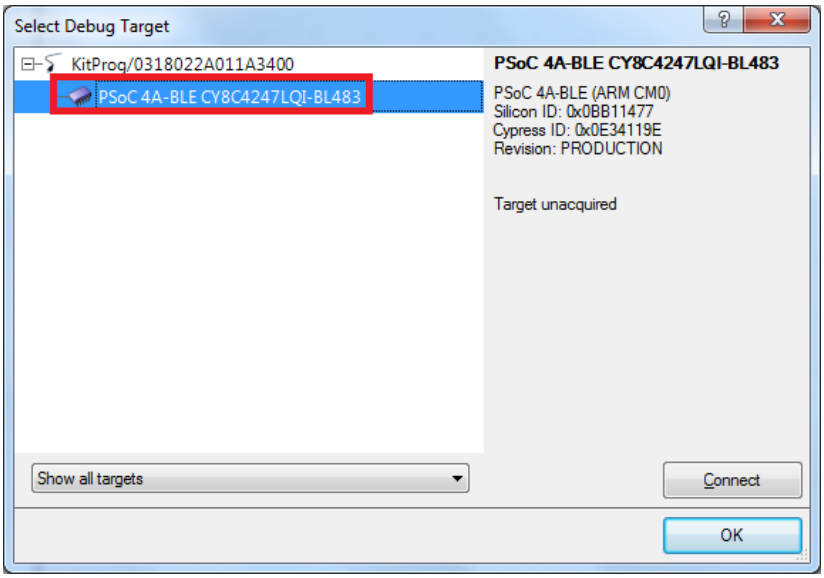
3. Select **Debug** > **Select Debug Target**, as shown below.

Figure 4. Selecting Debug Target



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

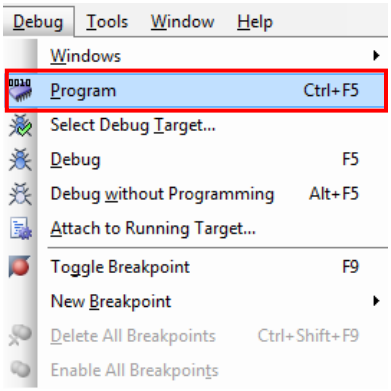
Figure 5. Selecting Debug Target



If you are using your own hardware, make sure the Port Setting configuration under Select Debug Target window for your programming hardware is configured as per your setup.

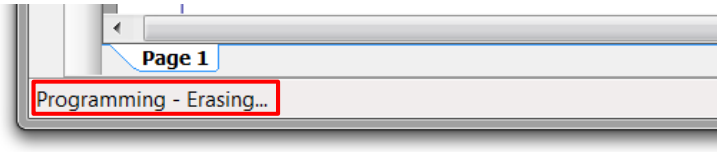
5. Select **Debug** > **Program** to program the device with the project, as shown below.

Figure 6. Programming Device



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

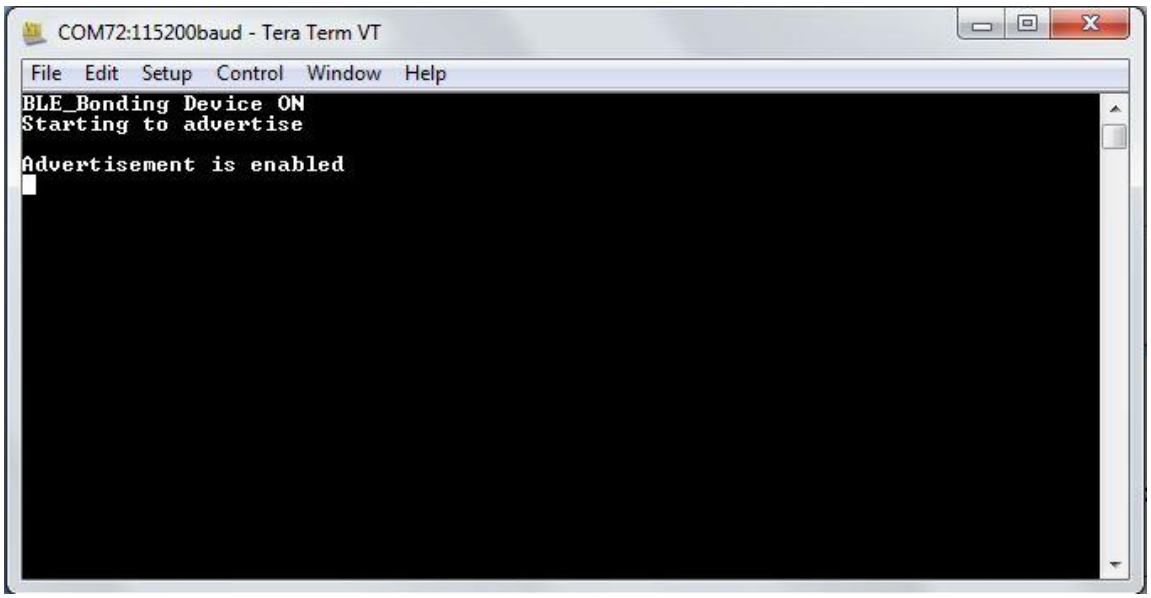
Figure 7. Programming status



Operation and Testing

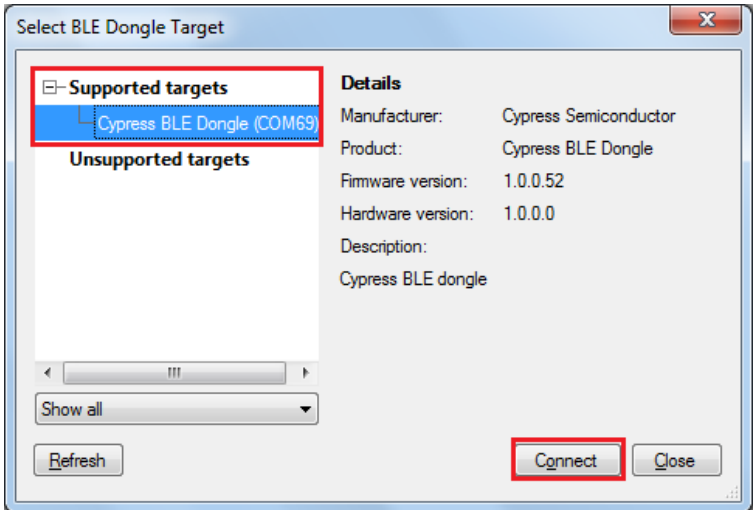
1) Having the kit connected and programmed, open a Serial terminal and have the BaudRate as 115,200. Press the Reset switch (SW1) on the Kit. You can see the UART messages to confirm if the terminal works.

Figure 8. Serial Terminal



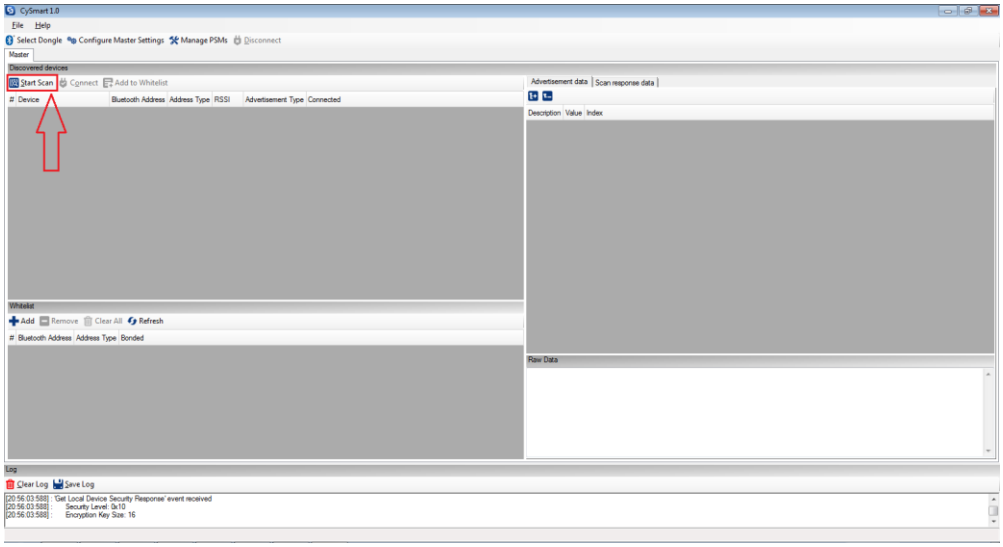
2) On your computer, launch **CySmart 1.0**. It is located in the **All Programs -> Cypress -> CySmart** folder in the Windows start menu. The tool opens up and asks you to **Select BLE Dongle Target**. Select the **Cypress BLE Dongle (COMxx)** and click **Connect**, as shown in below.

Figure 9. CySmart: Select BLE Dongle Target



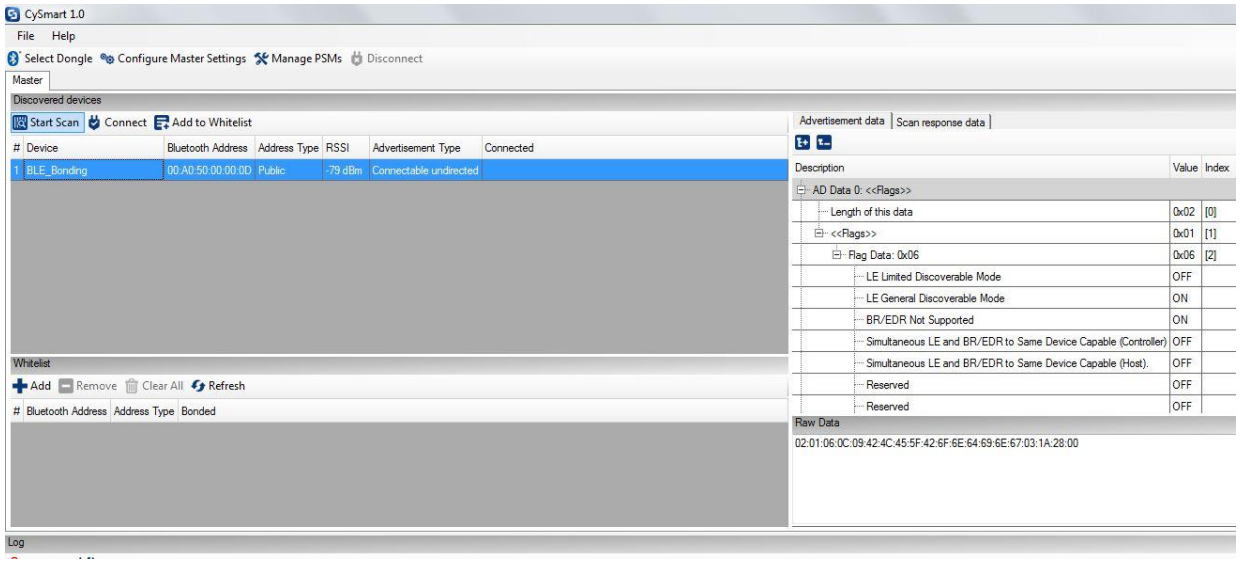
3) When the BLE-USB Bridge is connected, click on **Start Scan** to find your BLE device as shown in Figure 10.

Figure 10. CySmart scanning for devices



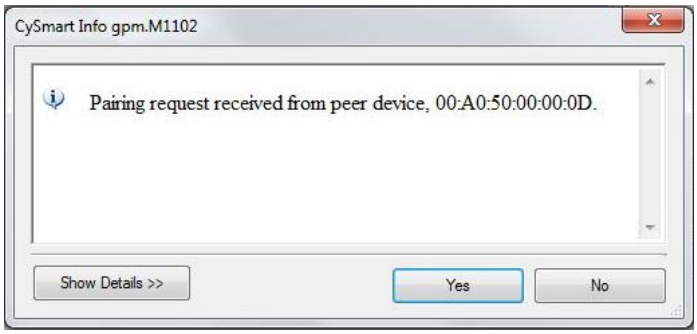
4) You can see the **BLE_Bonding** device detected.

Figure 11. CySmart listing available devices



5) If you press **Connect,** pairing request message appears in the dialogue box, as shown in Figure 12.

Figure 12. Pairing request on CySmart

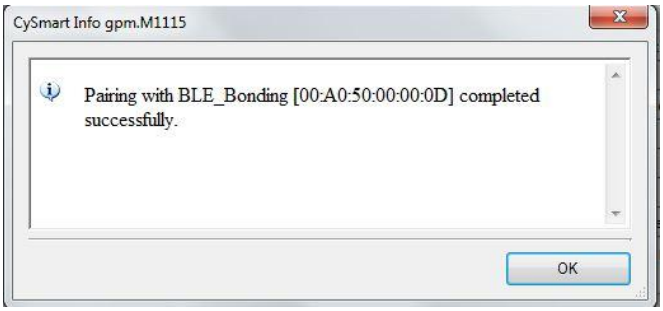


6) Click **Yes**. Now it will ask for the PassKey. The Passkey is displayed in the UART Terminal. Enter the passkey and press OK.

Figure 13. Console log for the project

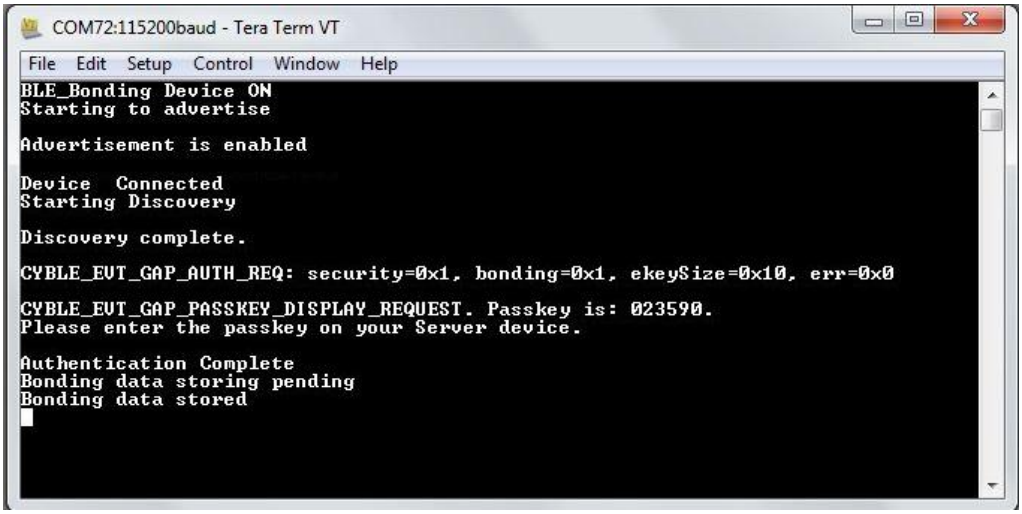
7) Upon successful pairing, the following message appears:

Figure 14. Successful pairing



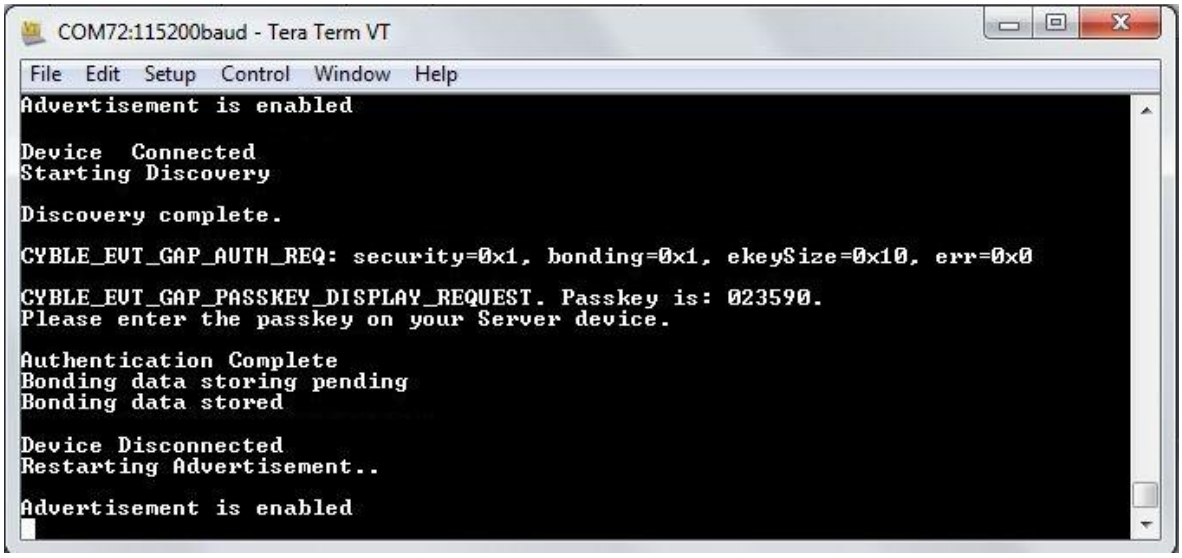
8) After Pairing, the device is automatically bonded and the corresponding message appears in the terminal, as shown in figure 15.

Figure 15. Bonding Complete Message



9) Now for disconnecting, press the disconnect button in CySmart. Once it is disconnected the disconnected, the corresponding message appears in the serial terminal as shown in Figure 16.

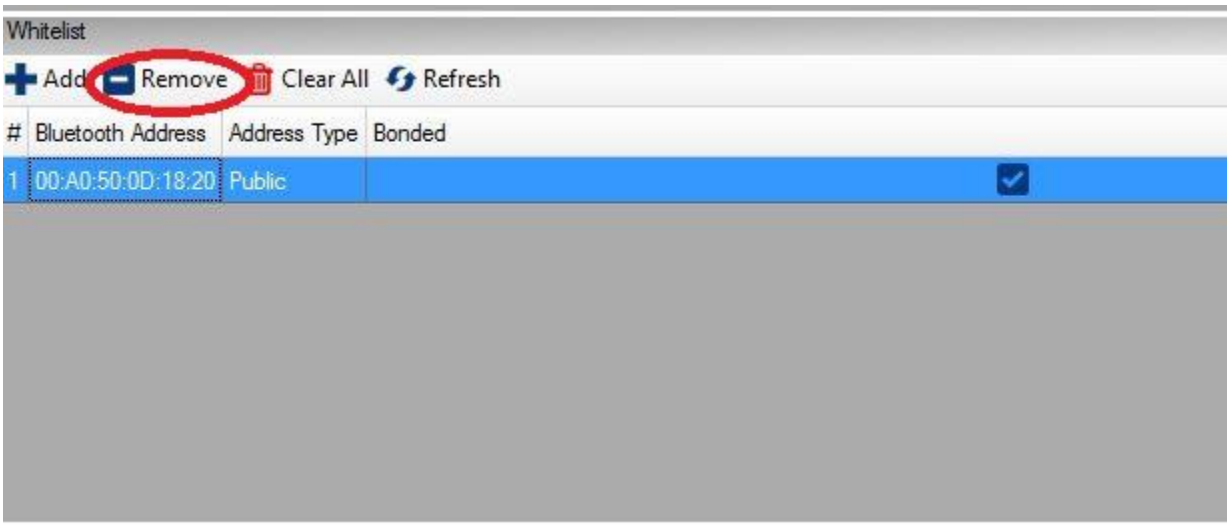
Figure 16. Disconnection message in Terminal



10) Press **Start Scan** again in CySmart and connect once again with the **BLE_Bonding** device. In this step as well, the message box related to pairing will be displayed, as shown in Figure 12. Click **OK**. This time user does not have to enter the passkey because the devices are already bonded.

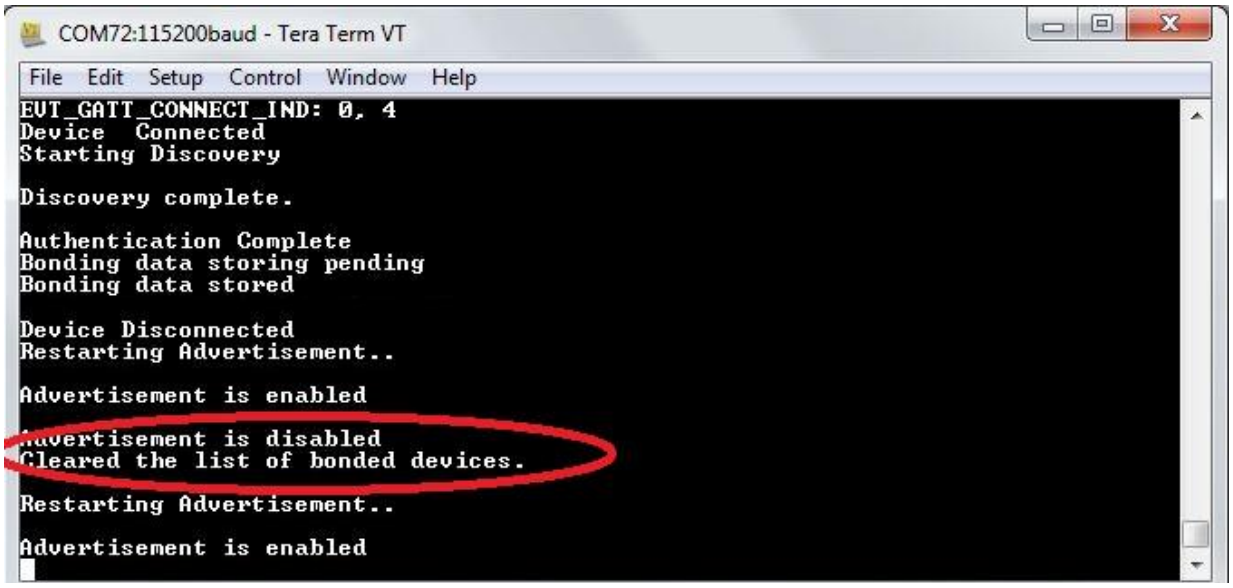
11) To remove the stored bonding data, disconnect the connection. The data has to be removed from both BLE-USB bridge and the BLE_Bonding device. In the **Whitelist** section in the CySmart, select the device and click remove to remove the device from the BLE-USB bridge.

Figure 17. Remove Bonding Information in CySmart



12) In the Serial terminal also, enter ‘R’. This would remove the bonding information and the related message appears as shown in figure 18.

Figure 18. Bonding Information Clear



13) As the bonding information has now been removed from both the devices, the authentication key has to be entered again for pairing up next time.

State of LEDs:

GREEN LED	Indicates that the BLE Component is currently advertising
RED LED	Indicates that the BLE Component is Idle (neither advertising nor connected)
BLUE LED	Indicates that the Authentication is complete

Related Documents

Table 1 lists all relevant application notes, code examples, knowledge base articles, device datasheets, and Component / user module datasheets.

Table 1. Related Documents

Document	Title	Comment
AN91267	Getting Started with PSoC® 4 BLE	Provides an introduction to PSoC 4 BLE device that integrates a Bluetooth Low Energy radio system along with programmable analog and digital resources.
AN91445	Antenna Design Guide	Provides guidelines on how to design an antenna for BLE applications.
001-99492	PSoC® 4: PSoC 4XX7_BLE Family Datasheet	Datasheet for the PSoC 4 BLE family of devices