

Objective

This example demonstrates the BLE Privacy feature using the PSoC 4 BLE family of devices.

Overview

This example uses the BLE Pioneer Kit to implement the BLE Privacy feature. The BLE Privacy feature provides higher level of security to the connection over a period of time. Once two devices have exchanged keys as part of the pairing process and have stored their keys (called bonding), they can use private addresses instead of public addresses to secure their connection. If a device has the encryption keys, it can resolve that private address and establish the identity of this device.

This example contains two projects – one for each for the GAP Central and GAP Peripheral sides of the BLE connection. The GAP Peripheral implements BLE Privacy by advertising with a resolvable private address, once it has bonded with the GAP Central device. The GAP Central resolves the GAP Peripheral's private address and then connects to it the next time. The two devices need not exchange encryption keys again, since the information is already present as part of Bonding.

The GAP Peripheral is implemented using the PSoC 4 BLE module on the BLE Pioneer Kit. The GAP Central is implemented using the CySmart USB Dongle in the same kit.

For more details on the BLE Privacy feature, refer to the Bluetooth 4.1 Specification, Volume 3, Part C, Section 10.7.

Requirements

Design Tool: [PSoC Creator 3.1 SP1](#)

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](#)

Hardware Setup

The BLE Pioneer Kit has all of the necessary hardware required for this lab. There is no special setup required.

PSoC Creator Schematic

Figure 1. PSoC Creator Schematic – GAP Peripheral

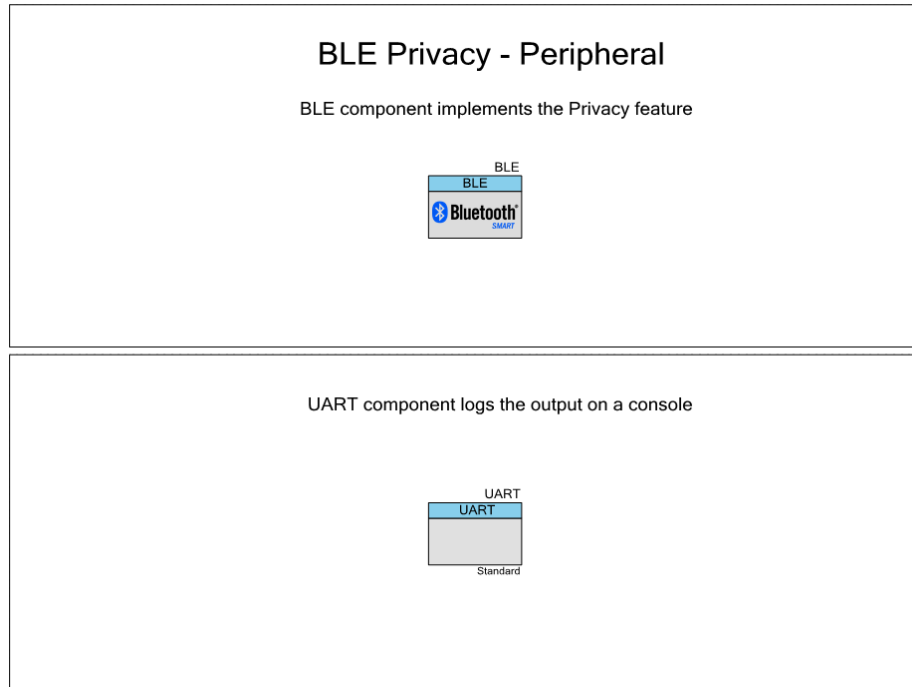
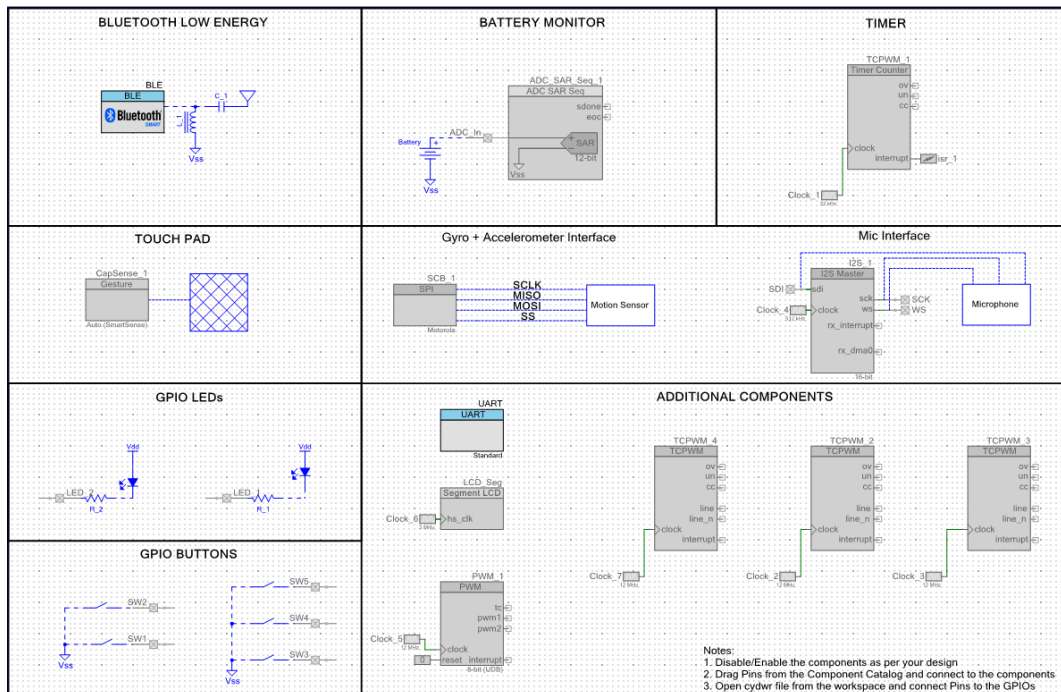


Figure 2. PSoC Creator Schematic – GAP Central



Project Description – GAP Peripheral

On the first time, the GAP Peripheral advertises with a public address. Once it is connected to a GAP Central device, the GAP Peripheral initiates the authentication request and the key exchange happens between the GAP Central and GAP Peripheral devices. After the key exchange and authentication is complete, the GAP Peripheral stores the peer device's information in flash memory (which is called Bonding).

After Bonding is complete and the GAP Peripheral is disconnected, it starts advertising with a random address, which is a resolvable private address. This address can be resolved by a GAP Central device if it has the required keys (specifically, the Identity Resolving Key or IRK). Only a GAP Central device which has the corresponding IRK can successfully resolve the private address of the GAP Peripheral and establish that this GAP Peripheral is the same as the one which it had connected to earlier. Thus, only a previously bonded GAP Central device can decipher this GAP Peripheral device.

A new GAP Central device can also connect to this GAP Peripheral (which advertises with a private address) without resolving its address, but then it would require a fresh authentication procedure. Thus, the connection between this GAP Peripheral and the original GAP Central device can be kept private.

The user can clear the GAP Peripheral's bonded device list by pressing 'R'. Once the bonded device list is cleared, authentication on the next connection with the same GAP Central device would require key exchange. For this to happen, the GAP Central device should not have this GAP Peripheral as part of its bonded device list. So the user should clear the GAP Central's list as well.

Project Description – GAP Central

The GAP Central device scans for all nearby devices (limited to 10 devices) and lists the devices on the UART console. The user can then connect to any device he wants by pressing 'C' followed by the device number from the list. Once connected to a GAP Peripheral, the GAP Central responds to the incoming authentication request and generates a pairing key, which the user has to enter on the GAP Peripheral's side. Once the keys are exchanged and authentication is complete, the GAP Central also stores the peer device's information (bonding).

Once the Bonding is complete, the user can press 'D' to disconnect from the GAP Peripheral. At that time, the GAP Central starts scanning for nearby devices and lists them. If it gets a random address and has the IRK, it tries to resolve the device address and shows the results (success / failure). For the random address which is resolved, the GAP Central device updates its bonding device list to change the device address from the old address to the new address. When connected to a GAP Peripheral with its private address resolved, the key exchange won't need to happen again since the device is already part of the GAP Central's bonded device list.

The user can press 'S' at any time during the scan process to refresh the scanned device list.

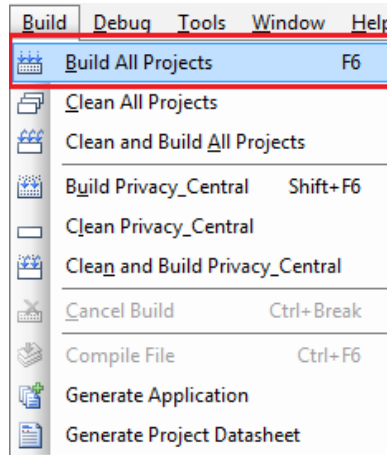
The user can press 'R' at any time to clear the bonded device list on the GAP Central. Once this list is cleared, the GAP Peripheral side's bonded device list should also be cleared in order to connect to it and ensure correct authentication the next time.

Build and Program

To work with this example, follow these steps –

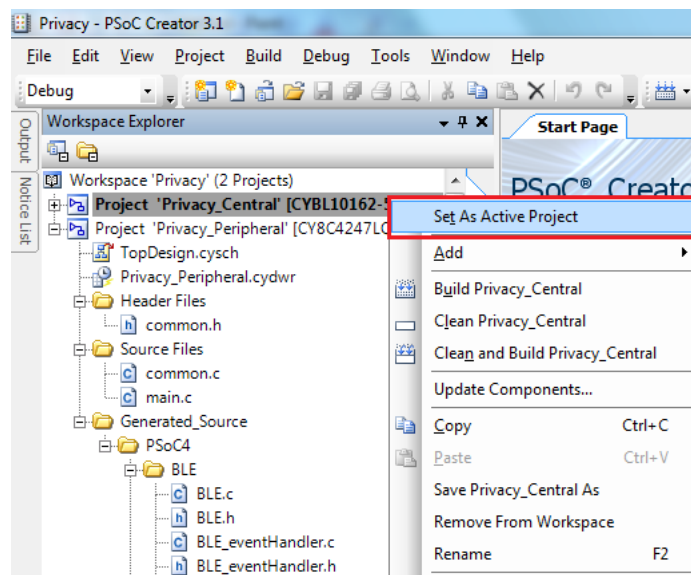
1. Open the workspace **Privacy** in PSoC Creator.
2. In PSoC Creator, select **Build > Build All Projects**, as shown in [Figure 3](#).

Figure 3. Build All Projects



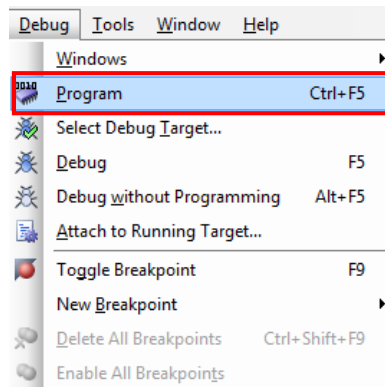
3. Programming the GAP Central: Right-click on **Privacy_Central** project and click **Set as Active Project**, as shown in Figure 4.

Figure 4. Selecting Central as the Active project



4. Plug the CySmart USB Dongle into your computer and Select **Debug > Program** to program the device with the project, as shown in Figure 5.

Figure 5. Programming the Device



5. Programming the GAP Peripheral: Now set the **Privacy_Peripheral** as the active project. Right click on the project name and select **Set as Active Project**.
6. Plug in the BLE Pioneer Kit with the PSoC 4 BLE module and select **Debug > Program** to program the kit.

Testing

Follow these steps to test the example –

1. Open a terminal emulator such as Putty or Tera Term for both the GAP Central (CySmart USB Dongle) and the GAP Peripheral (BLE Pioneer Kit) devices. The COM settings are: Baud rate – 115200 bps, Data bits – 8, Stop bits – 1, Parity – None.
2. Reset both the devices to see this output shown in [Figure 6](#) and [Figure 7](#) on the terminals.

Figure 6. Terminal output – GAP Peripheral

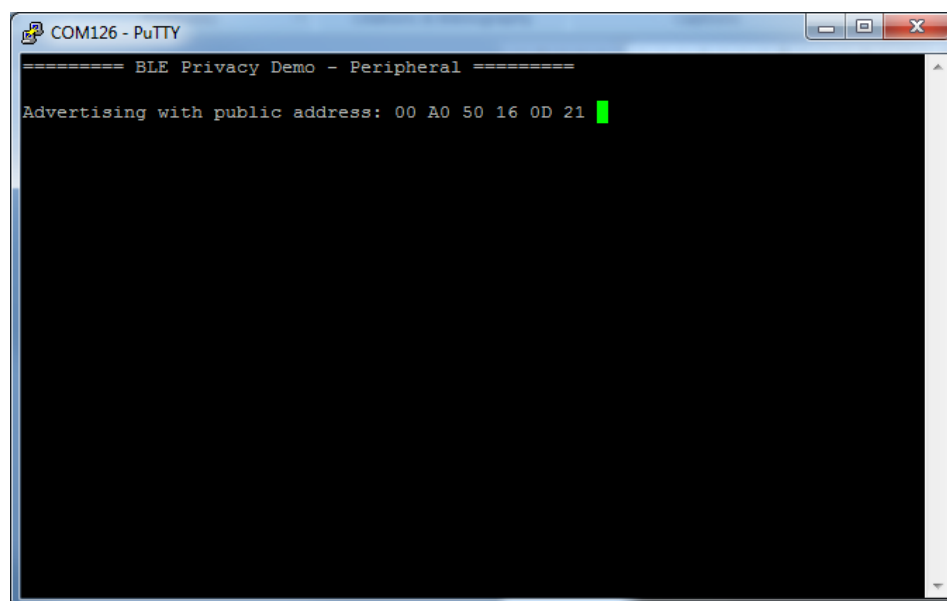
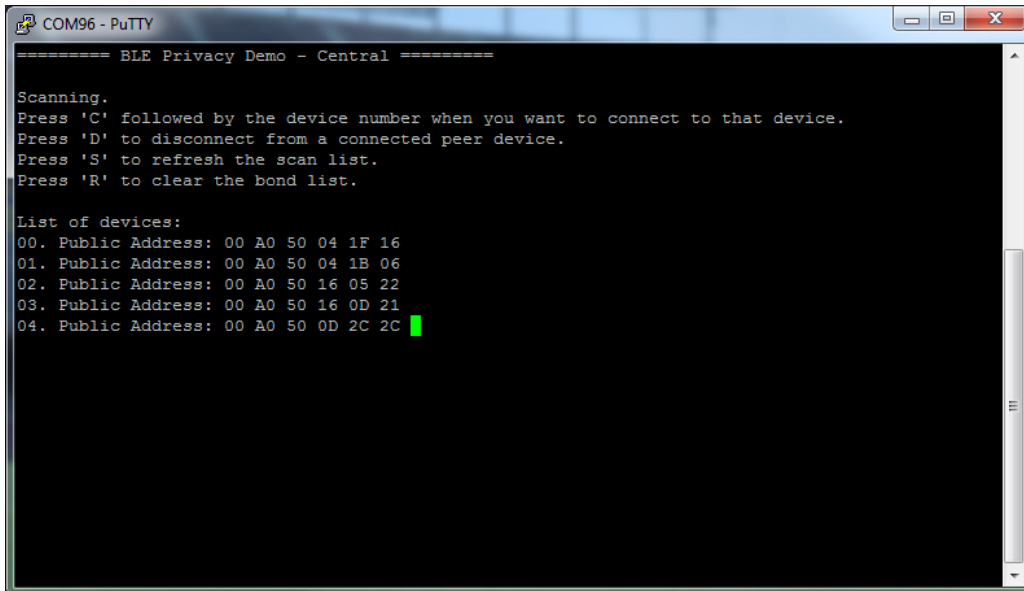


Figure 7. Terminal output – GAP Central



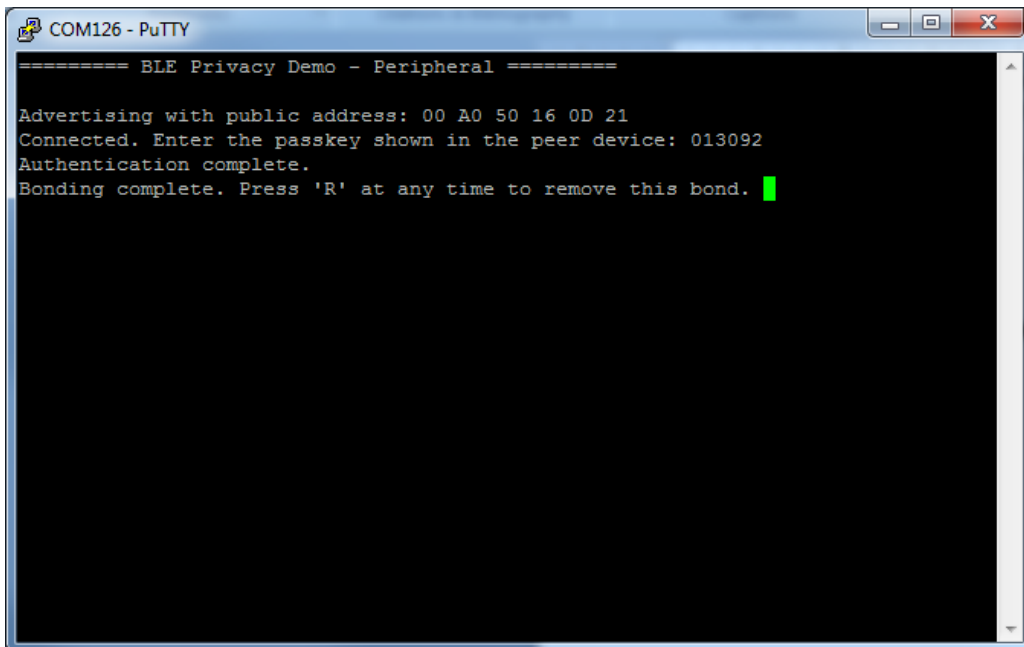
```
COM96 - PuTTY
===== BLE Privacy Demo - Central =====

Scanning.
Press 'C' followed by the device number when you want to connect to that device.
Press 'D' to disconnect from a connected peer device.
Press 'S' to refresh the scan list.
Press 'R' to clear the bond list.

List of devices:
00. Public Address: 00 A0 50 04 1F 16
01. Public Address: 00 A0 50 04 1B 06
02. Public Address: 00 A0 50 16 05 22
03. Public Address: 00 A0 50 16 0D 21
04. Public Address: 00 A0 50 0D 2C 2C
```

3. On the GAP Central terminal, press 'C' followed by the device number for the GAP Peripheral's address. The authentication procedure will then start and you will be asked to enter the passkey in the GAP Peripheral terminal as shown in [Figure 8](#) and [Figure 9](#).

Figure 8. Connection and Authentication – GAP Peripheral



```
COM126 - PuTTY
===== BLE Privacy Demo - Peripheral =====

Advertising with public address: 00 A0 50 16 0D 21
Connected. Enter the passkey shown in the peer device: 013092
Authentication complete.
Bonding complete. Press 'R' at any time to remove this bond.
```

Figure 9. Connection and Authentication – GAP Central

```

COM96 - PuTTY
===== BLE Privacy Demo - Central =====

Scanning.
Press 'C' followed by the device number when you want to connect to that device.
Press 'D' to disconnect from a connected peer device.
Press 'S' to refresh the scan list.
Press 'R' to clear the bond list.

List of devices:
00. Public Address: 00 A0 50 04 1F 16
01. Public Address: 00 A0 50 04 1B 06
02. Public Address: 00 A0 50 16 05 22
03. Public Address: 00 A0 50 16 0D 21
04. Public Address: 00 A0 50 0D 2C 2C
Connect to device: 3
Connected. Enter this passkey in your peer device: 013092
Authentication complete.
Bonding complete. █

```

4. Once the devices are connected, you can press 'D' on the GAP Central terminal to disconnect from the GAP Peripheral. As soon as you disconnect, the GAP Peripheral will start advertising with a private address using the IRK generated as part of the authentication procedure just finished. The GAP Central will list this private address, showing that it can resolve this address as well. See [Figure 10](#) and [Figure 11](#).

Figure 10. GAP Peripheral Advertising with Private address

```

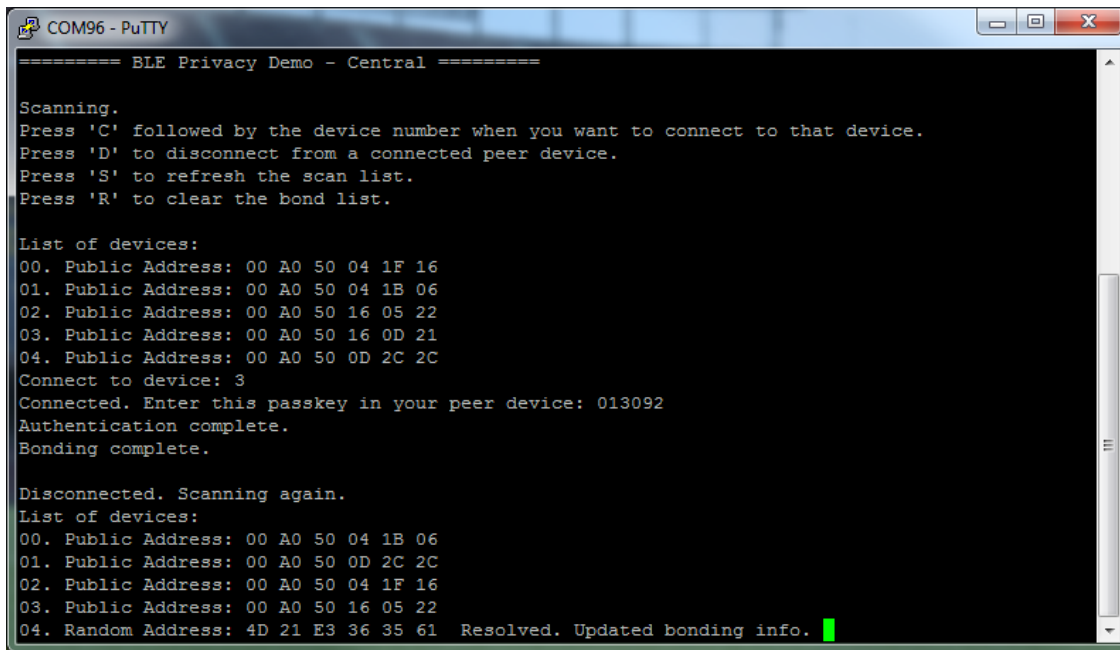
COM126 - PuTTY
===== BLE Privacy Demo - Peripheral =====

Advertising with public address: 00 A0 50 16 0D 21
Connected. Enter the passkey shown in the peer device: 013092
Authentication complete.
Bonding complete. Press 'R' at any time to remove this bond.

Disconnected. Advertising with new private address: 4D 21 E3 36 35 61 █

```

Figure 11. GAP Central listing nearby devices with resolution status for private addresses



```

===== BLE Privacy Demo - Central =====

Scanning.
Press 'C' followed by the device number when you want to connect to that device.
Press 'D' to disconnect from a connected peer device.
Press 'S' to refresh the scan list.
Press 'R' to clear the bond list.

List of devices:
00. Public Address: 00 A0 50 04 1F 16
01. Public Address: 00 A0 50 04 1B 06
02. Public Address: 00 A0 50 16 05 22
03. Public Address: 00 A0 50 16 0D 21
04. Public Address: 00 A0 50 0D 2C 2C
Connect to device: 3
Connected. Enter this passkey in your peer device: 013092
Authentication complete.
Bonding complete.

Disconnected. Scanning again.
List of devices:
00. Public Address: 00 A0 50 04 1B 06
01. Public Address: 00 A0 50 0D 2C 2C
02. Public Address: 00 A0 50 04 1F 16
03. Public Address: 00 A0 50 16 05 22
04. Random Address: 4D 21 E3 36 35 61 Resolved. Updated bonding info.

```

5. Every 60 seconds, the GAP Peripheral will restart advertisement with a new private address and the GAP Central device will list this new address in its scan list. You can press 'S' at any time to refresh scan.
6. Connect to the GAP Peripheral with the private address now. The bonding should now happen automatically.

Restoring the CySmart USB Dongle Firmware

To restore to the original functionality of the CySmart USB Dongle, locate the Dongle hex file in the Kit installation directory. A typical installation path is:

C:\Program Files\Cypress\CY8CKIT-042-BLE Kit\1.0\Firmware\BLE Dongle\Hex Files\BLE_Dongle_CySmart.hex

Open PSoC Programmer and load this file in the tool. Then connect to the KitProg of the CySmart USB Dongle in PSoC Programmer and program the hex file. This will restore the original functionality of the CySmart USB Dongle.

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.