PSoC 4 BLE – Peripheral and Broadcaster

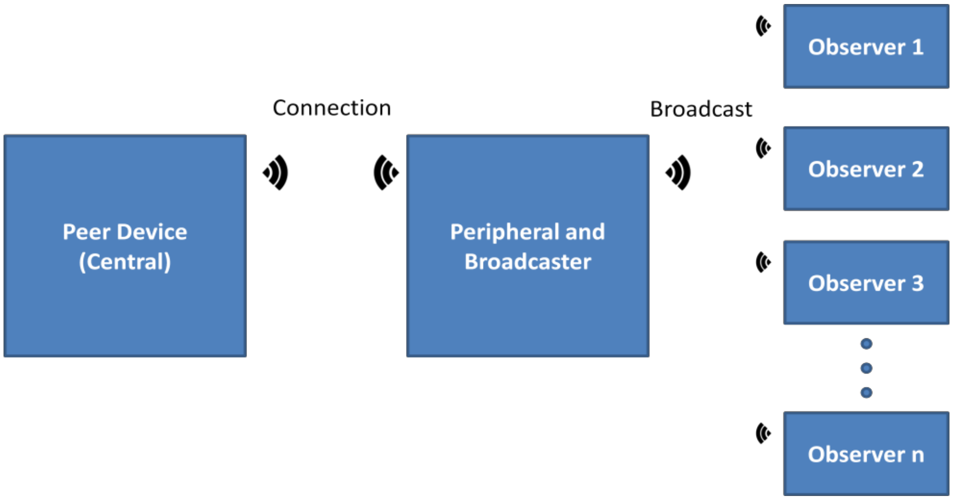
Objective

This example demonstrates functionality where the device is a Peripheral in an active connection and is also a Broadcaster at the same time.

Overview

This example uses the BLE Pioneer Kit to implement a custom profile with the GAP Peripheral role in a device which enters a connection with a Central device. As soon as the connection is created, the Peripheral device also starts broadcasting so that it can be seen by other devices in the vicinity.

Figure 1. Peripheral and Broadcaster roles



Requirements

Design Tool: PSoC Creator 4.0 Update 1, CySmart 1.2

Programming Language: C (GCC 4.9 – included with PSoC Creator)

Associated Devices: All PSoC 4 BLE devices

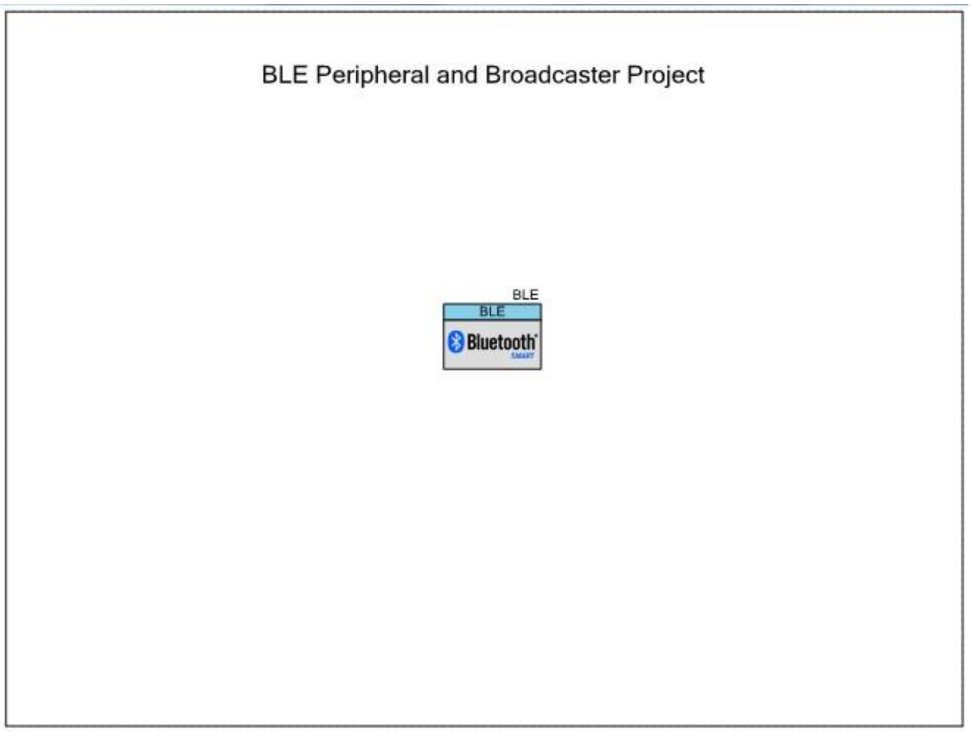
Required Hardware: CY8CKIT-042-BLE Bluetooth® Low Energy (BLE) Pioneer Kit

Hardware Setup

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

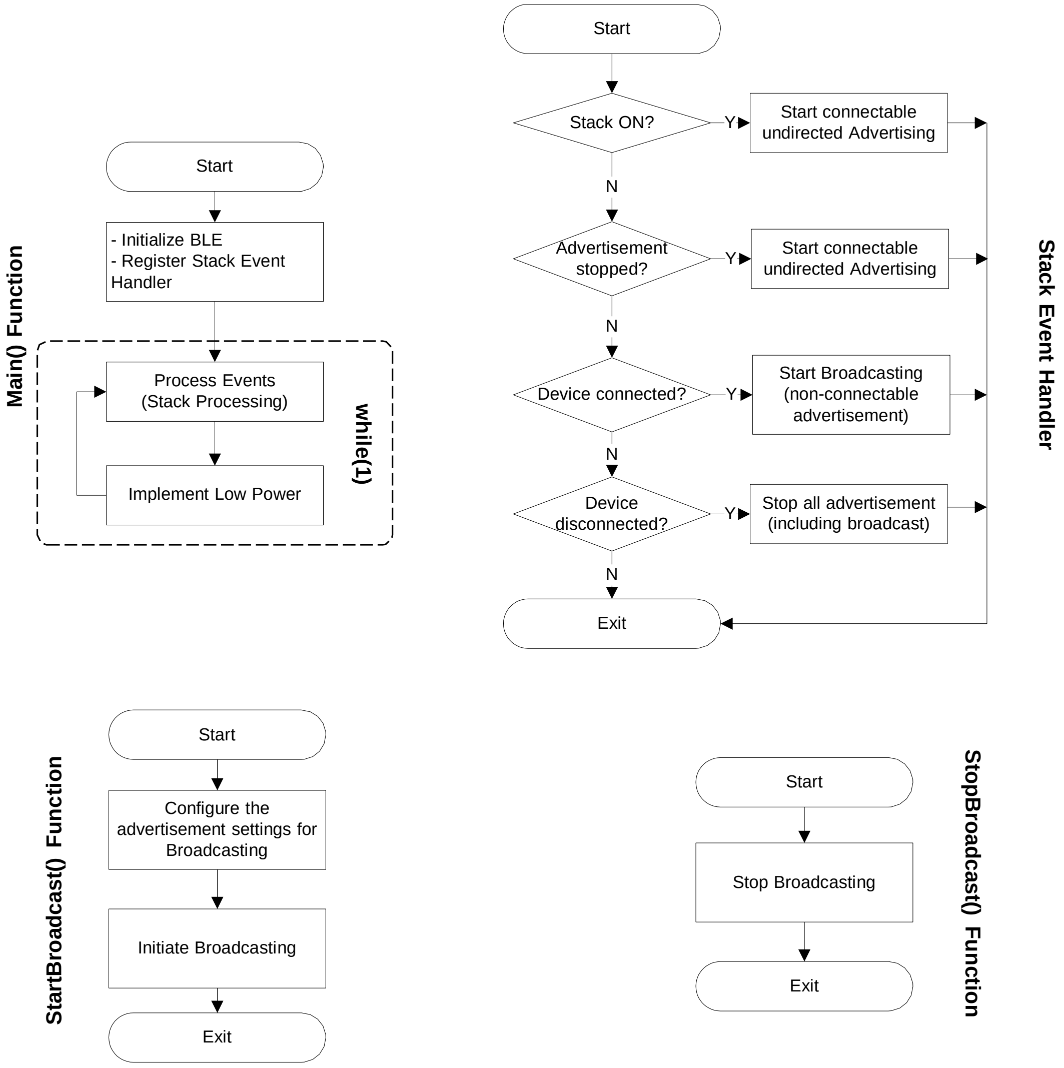
PSoC Creator Schematic

Figure 2. PSoC Creator Schematic



Firmware Flow

Figure 3. Firmware Flow


Start
Stack ON?
Y
Start connectable undirected Advertising
N
Advertisement stopped?
Y
Start connectable undirected Advertising
N
Device connected?
Y
Start Broadcasting (non-connectable advertisement)
N
Device disconnected?
Y
Stop all advertisement (including broadcast)
N
Exit
Stack Event Handler
Main() Function
Start
- Initialize BLE
- Register Stack Event Handler
Process Events (Stack Processing)
Implement Low Power
while(1)
StartBroadcast() Function
Start
Configure the advertisement settings for Broadcasting
Initiate Broadcasting
Exit
StopBroadcast() Function
Start
Stop Broadcasting
Exit

1. **main() function**: This is the central function which performs the initialization of the BLE Stack. It then executes the necessary routines to process the BLE events and maintain the connection. It also implements low power in the system by first requesting BLE to enter deep sleep and then putting the system to deep sleep.
   In the initial section of the *main()* function, the API function *CyBle_Start(StackEventHandler)* is called to start the BLE Component and register a callback to the Stack event handler. Note that the callback function can have any name – in this project, we used StackEventHandler. Once the system is initialized, *main()* continuously operates in a *while(1)* loop executing *CyBle_ProcessEvents()* and system low power implementation. This function processes the events received by the BLE Stack and enables the application layer to use them and take the appropriate action.

2. **StackEventHandler() function**: This function handles the common events generated for the BLE Stack. For example, the event *CYBLE_EVT_STACK_ON* is received when the Stack is initialized and turned ON. The event *CYBLE_EVT_GAP_DEVICE_DISCONNECTED* is received when the BLE connection is disconnected.

3. **StartBroadcast() function**: This function first configures the advertisement settings for broadcast, and then starts the broadcast process. It configures the following settings:

   Advertisement Type – Non-connectable undirected advertisement

   Advertisement BD address – Public

   Advertisement interval – 100 ms

   Advertisement channel map – All channels (37, 38, 39)

   Advertisement Data – Advertisement Flags and the Device Name. The device name is “Peripheral and Broadcaster” when it is not connected (thus acting as a Peripheral) and “Broadcaster only” when it is connected.

   Scan Response Data – None

4. **StopBroadcast() function:** This function stops the broadcast process.

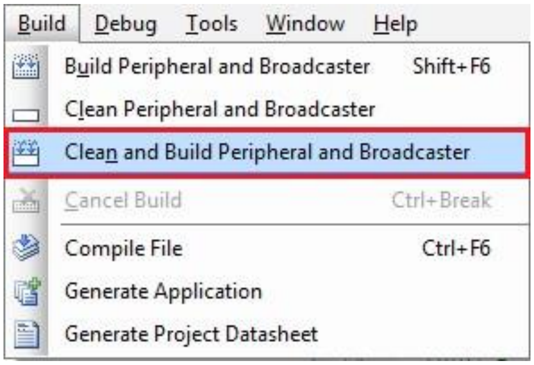
Build and Program

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.

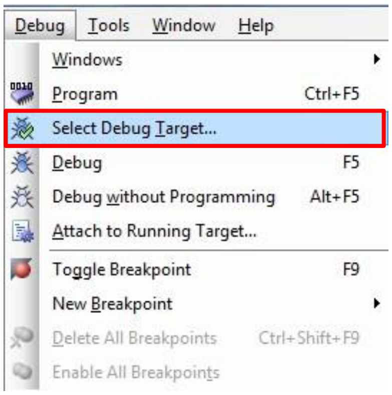
1. On PSoC Creator, select **Build** > **Clean and Build Peripheral and Broadcaster**, as shown in Figure 4.

Figure 4. Build Project


Build
Debug
Tools
Window
Help
Build Peripheral and Broadcaster Shift+F6
Clean Peripheral and Broadcaster
Clean and Build Peripheral and Broadcaster
Cancel Build Ctrl+Break
Compile File Ctrl+F6
Generate Application
Generate Project Datasheet

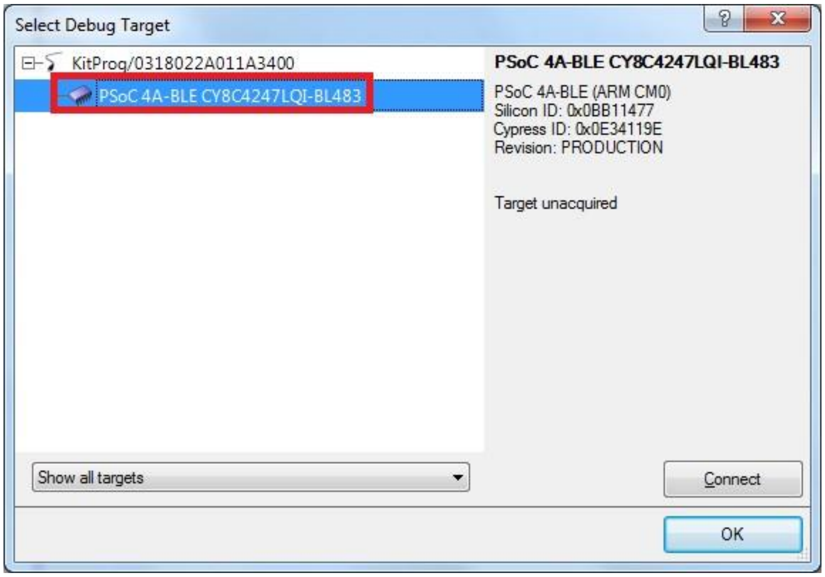
2. Select **Debug** > **Select Debug Target**, as shown in Figure 5.

Figure 5. Selecting Debug Target



3. In the **Select Debug Target** dialog box, click **Port Acquire**, and then click **Connect** as shown in Figure 6. Click **OK** to close the dialog box.

Figure 6. Connecting to a Device



Select Debug Target

KitProg/0318022A011A3400

PSoC 4A-BLE CY8C4247LQI-BL483

PSoC 4A-BLE CY8C4247LQI-BL483

PSoC 4A-BLE (ARM CM0)
Silicon ID: 0x0BB11477
Cypress ID: 0x0E34119E
Revision: PRODUCTION

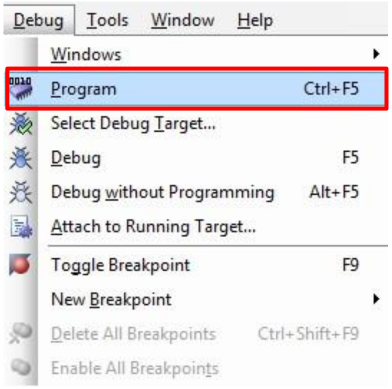
Target unacquired

Show all targets

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.

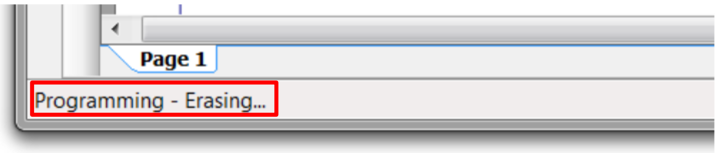
4. Select **Debug** > **Program** to program the device with the project, as shown in Figure 7.

Figure 7. Programming the Device



You can view the programming status on the PSoC Creator status bar (lower-left corner of the window), as shown in Figure 8.

Figure 8. Programming Status



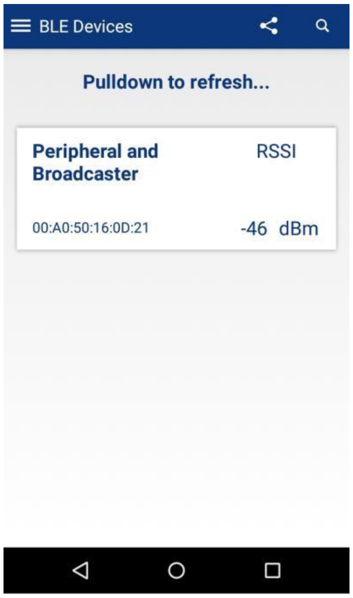
Programming - Erasing...

Testing

Testing with the CySmart BLE Test and Debug Utility iOS® or Android™ Mobile Apps:

1. Plug the BLE-USB Bridge (included with the BLE Pioneer Kit) in your computer's USB port.
2. On your BLE-enabled mobile phone, open the **CySmart app**.
3. Once the app is open, **swipe down** to refresh the list of nearby advertising BLE devices. See Figure 9.

Figure 9. CySmart App Scanning for BLE Devices



Pulldown to refresh...

Peripheral and Broadcaster

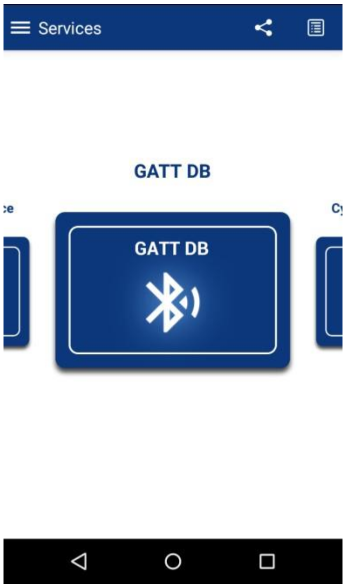
RSSI

00:A0:50:16:0D:21

-46 dBm

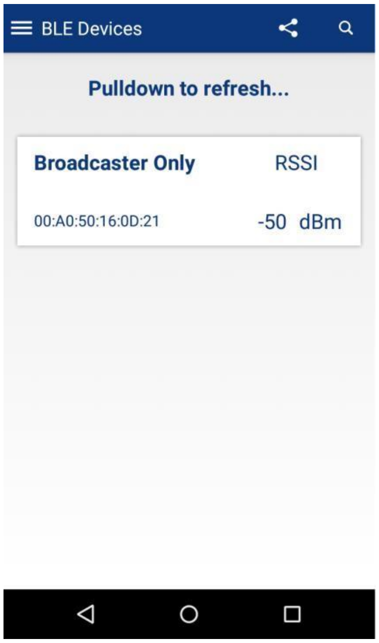
4. **Tap** on the **Peripheral and Broadcaster** device to connect to it. You will now see the Service tab showing the GATT DB on top, as shown in Figure 10.

Figure 10. CySmart App connected to the device



5. Now take another BLE-enabled mobile phone and open the **CySmart app** on it.
6. Notice that the app lists the device with the same address but the name is now changed to **Broadcaster only**. See Figure 11.

Figure 11. Device broadcasting during connection



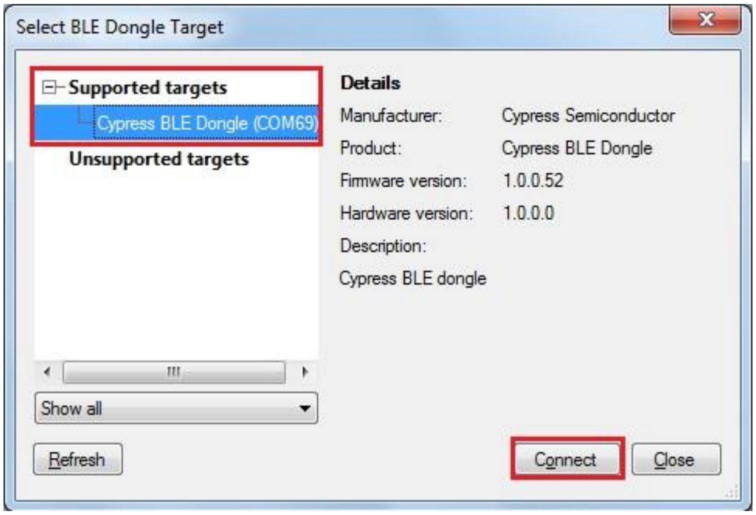
Broadcaster Only RSSI

00:A0:50:16:0D:21 -50 dBm

Testing with the CySmart BLE Test and Debug Utility for Windows PC:

1. Plug the BLE-USB Bridge (included with the BLE Pioneer Kit) in your computer's USB port.
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 Figure 12.

Figure 12: CySmart: Select BLE Dongle Target



Details

Manufacturer: Cypress Semiconductor

Product: Cypress BLE Dongle

Firmware version: 1.0.0.52

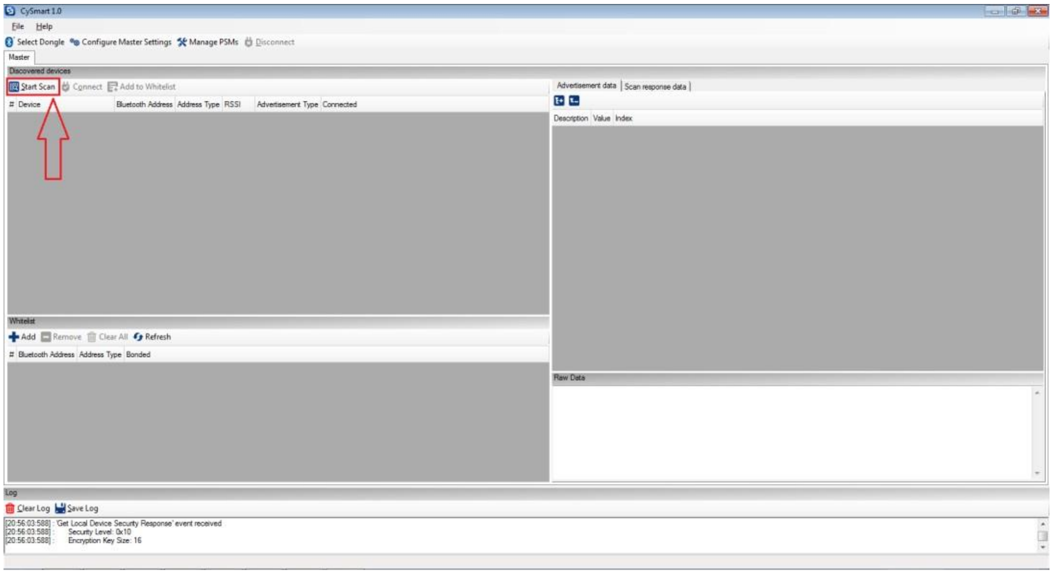
Hardware version: 1.0.0.0

Description:

Cypress BLE dongle

3. When the USB Dongle is connected, click on **Start Scan** to find your BLE device. See Figure 13.

Figure 13: Finding a BLE Device


CySmart 1.0
File Help
Select Dongle Configure Master Settings Manage PSMs Disconnect
Master
Discovered devices
Start Scan Connect Add to Whitelist
Advertisement data | Scan response data |
Whitelist
Add Remove Clear All Refresh
Raw Data
Log
Clear Log Save Log

#	Device	Bluetooth Address	Address Type	RSSI	Advertisement Type	Connected

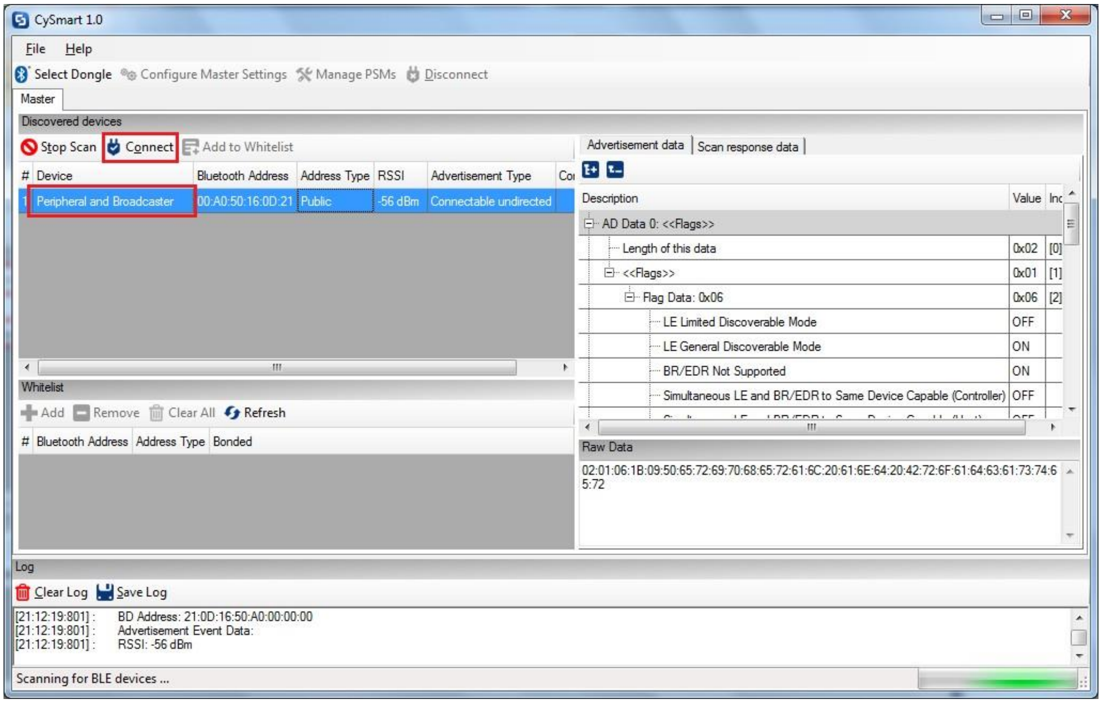
Description	Value	Index

#	Bluetooth Address	Address Type	Bonded

```
[20:56:03:588] : 'Get Local Device Security Response' event received
[20:56:03:588] :      Security Level: 0x10
[20:56:03:588] :      Encryption Key Size: 16
```

4. The tool lists all the nearby devices in the Discovered devices section. See that you get **Peripheral and Broadcaster** device in the list.
5. Connect to your device by double-clicking on the name of the device, or by selecting your device and clicking **Connect**. See Figure 14.

Figure 14: Detecting a Peripheral device


CySmart 1.0
File Help
Select Dongle Configure Master Settings Manage PSMs Disconnect
Master
Discovered devices
Stop Scan Connect Add to Whitelist
Advertisement data | Scan response data |
Whitelist
Add Remove Clear All Refresh
Raw Data
Log
Clear Log Save Log

#	Device	Bluetooth Address	Address Type	RSSI	Advertisement Type	Co
1	Peripheral and Broadcaster	00:A0:50:16:0D:21	Public	-56 dBm	Connectable undirected	

Description	Value	Inc
AD Data 0: <<Flags>>		
Length of this data	0x02	[0]
<<Flags>>	0x01	[1]
Flag Data: 0x06	0x06	[2]
LE Limited Discoverable Mode	OFF	
LE General Discoverable Mode	ON	
BR/EDR Not Supported	ON	
Simultaneous LE and BR/EDR to Same Device Capable (Controller)	OFF	

#	Bluetooth Address	Address Type	Bonded

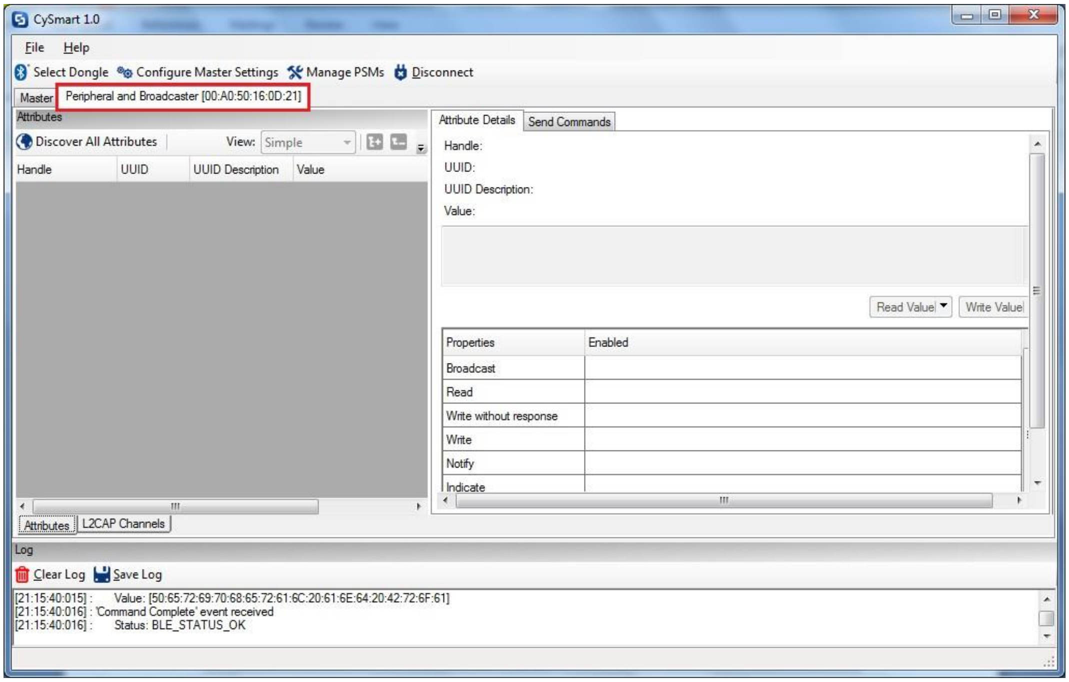
```
02:01:06:1B:09:50:65:72:69:70:68:65:72:61:6C:20:61:6E:64:20:42:72:6F:61:64:63:61:73:74:6
5:72
```

```
[21:12:19:801] :    BD Address: 21:0D:16:50:A0:00:00:00
[21:12:19:801] :    Advertisement Event Data:
[21:12:19:801] :    RSSI: -56 dBm
```

Scanning for BLE devices ...

6. Once the device is connected, the tool will open a separate tab for the device, shown in Figure 15.

Figure 15: CySmart connected to a BLE Device



```
[21:15:40:015] :      Value: [50:65:72:69:70:68:65:72:61:6C:20:61:6E:64:20:42:72:6F:61]
[21:15:40:016] : 'Command Complete' event received
[21:15:40:016] :      Status: BLE_STATUS_OK
```

7. Now open the CySmart app on a BLE-enabled phone and see that the app lists the same device with the name **Broadcast Only** as shown in Figure 11.

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.