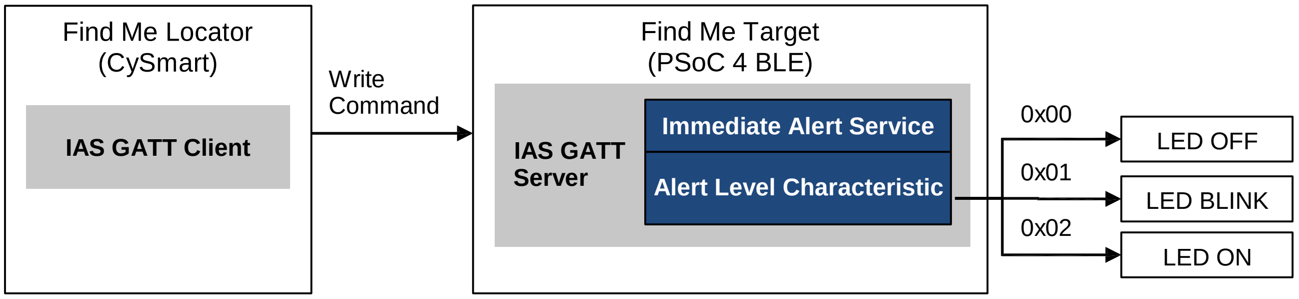
Objective

This example demonstrates the use of the BLE Component to design a simple Find Me application.

Overview

This example uses the BLE Pioneer Kit to design a simple Find Me application using the standard services defined by the Bluetooth SIG. In this example, the Find Me Profile instantiates the Immediate Alert Service which can be used to control the Alert Level of an indicator (e.g. buzzer, LED, etc.).

Figure 1: PSoC 4 BLE Find Me Application



Requirements

Design Tool: PSoC Creator 4.0 Update 1 , CySmart 1.2

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

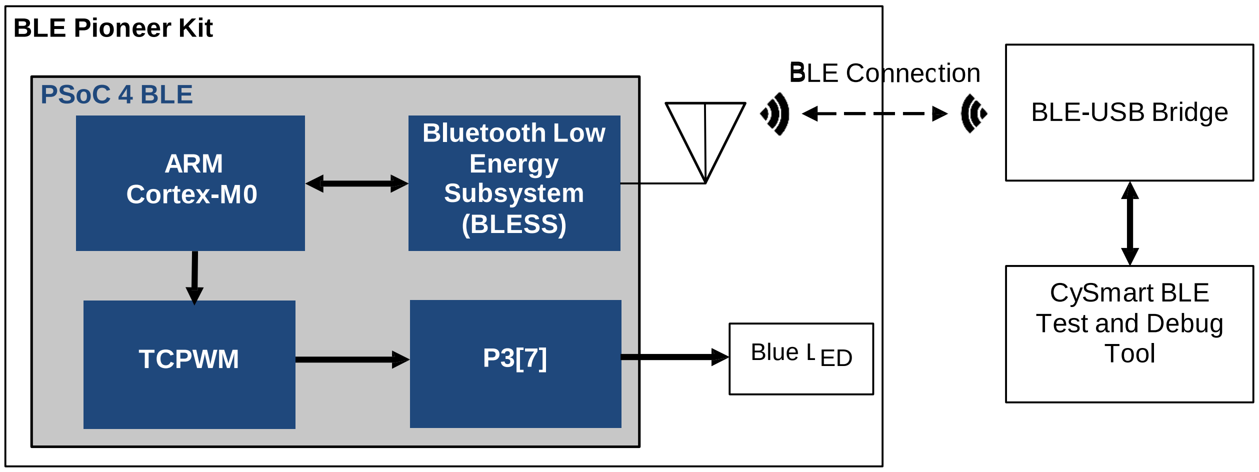
Associated Devices: All PSoC 4 BLE devices

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

Hardware Setup

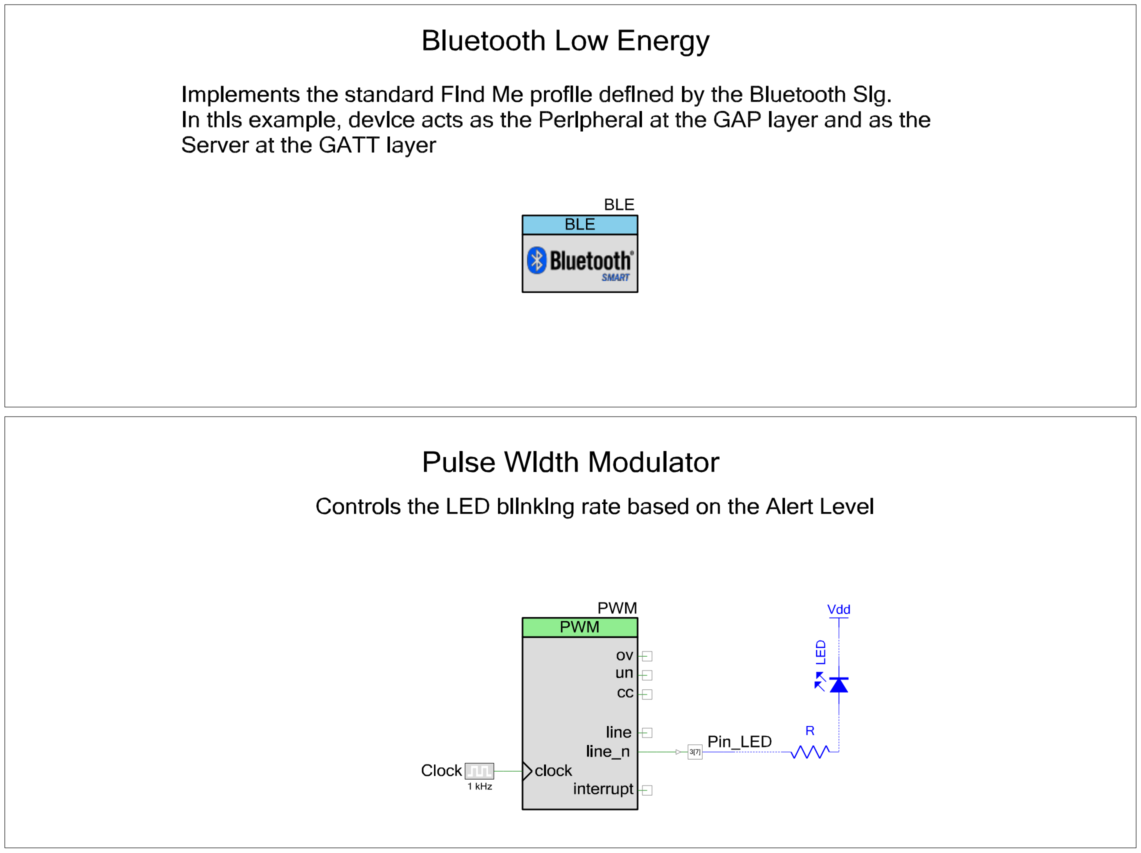
The BLE Pioneer Kit has all of the necessary hardware required for this lab. In this setup, the Blue LED is connected to pin P3.7 of the PSoC 4 BLE device.

Figure 2: Block Diagram



PSoC Creator Schematic

Figure 3. PSoC Creator Schematic



Bluetooth Low Energy

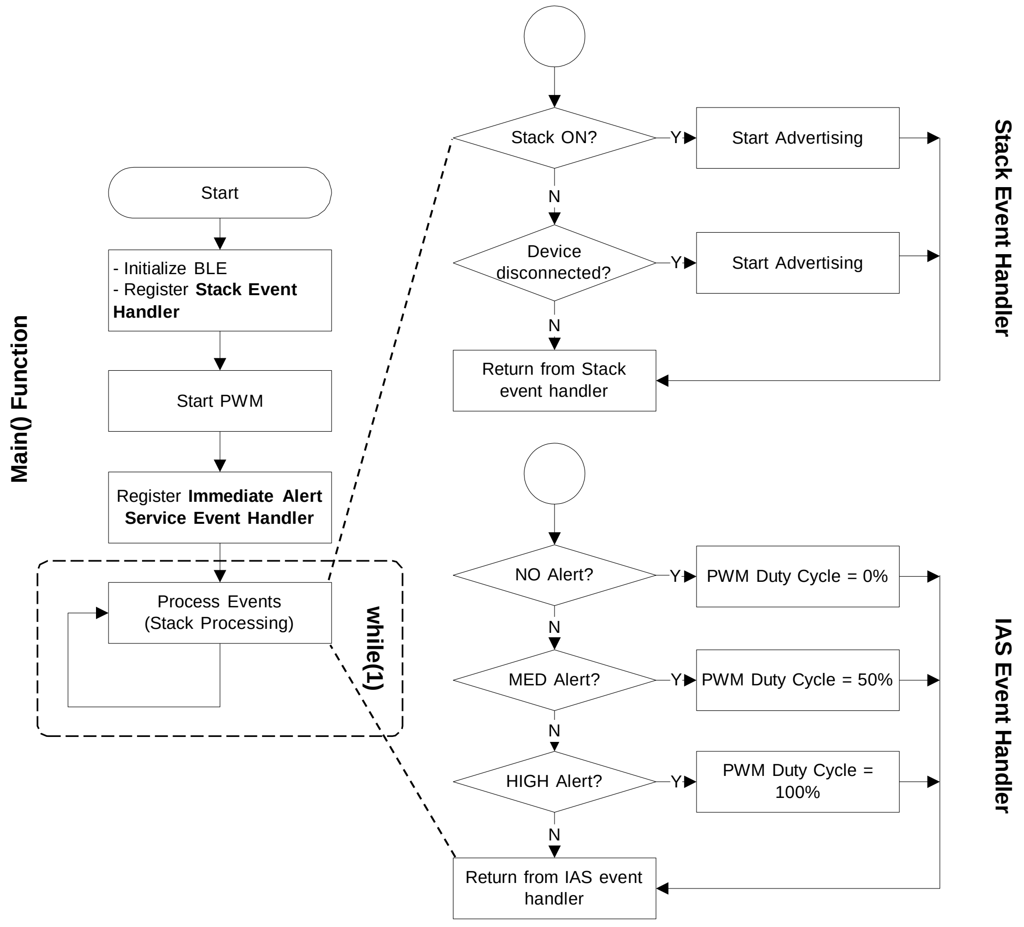
Implements the standard FInd Me proflle deflned by the Bluetooth SIg.
In thls example, devlce acts as the PerIpheral at the GAP layer and as the
Server at the GATT layer

Pulse WIdth Modulator

Controls the LED bIInkIng rate based on the AIert LeveI

Firmware Flow

Figure 4. Firmware Flow



1. **main() function**: This is the central function which performs the initialization of the BLE Stack and PWM for the LED control. It then executes the necessary routines to process the BLE events and maintain the connection.
   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()*. 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. **IasEventHandler() function**: This function handles the events for Immediate Alert Service. As a part of the event, it receives the alert levels which are used to drive the LED as per Table 1.

Table 1: Alert Level vs LED Blink Rate

Alert Level	PWM Duty Cycle	LED Status
NO_ALERT	100%	Always OFF
MILD_ALERT	50%	LED toggling every second
HIGH_ALERT	0%	Always ON

Note: LED pin is connected to the inverted terminal (line_n) of PWM, thus 100% duty cycle of PWM corresponds to LED always OFF.

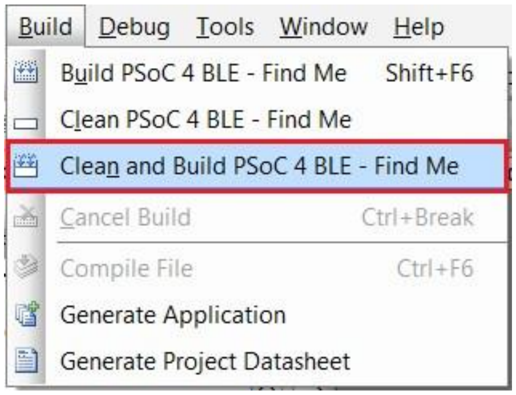
Build and Program

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.

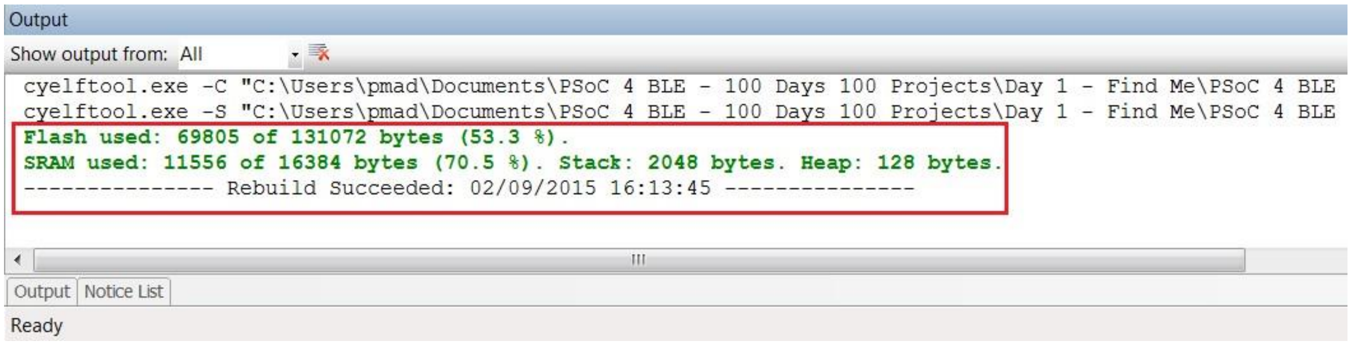
1. On PSoC Creator, select **Build** > **Clean and Build PSoC 4BLE Find Me**, as shown in Figure 5.

Figure 5. Build Project


Build
Debug
Tools
Window
Help
Build PSoC 4 BLE - Find Me   Shift+F6
Clean PSoC 4 BLE - Find Me
Clean and Build PSoC 4 BLE - Find Me
Cancel Build   Ctrl+Break
Compile File   Ctrl+F6
Generate Application
Generate Project Datasheet

2. On a successful build, the total flash and SRAM usage is reported as shown in Figure 6.

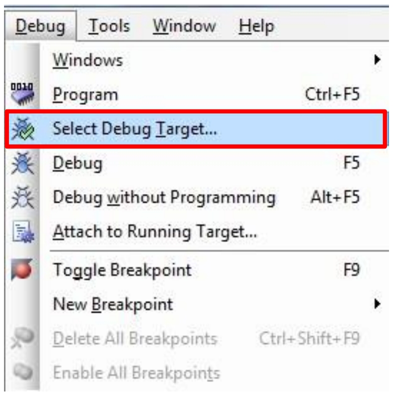
Figure 6. Build Succeeded



```
cyelftool.exe -C "C:\Users\pmad\Documents\PSoC 4 BLE - 100 Days 100 Projects\Day 1 - Find Me\PSoC 4 BLE
cyelftool.exe -S "C:\Users\pmad\Documents\PSoC 4 BLE - 100 Days 100 Projects\Day 1 - Find Me\PSoC 4 BLE
Flash used: 69805 of 131072 bytes (53.3 %).
SRAM used: 11556 of 16384 bytes (70.5 %). Stack: 2048 bytes. Heap: 128 bytes.
--------------- Rebuild Succeeded: 02/09/2015 16:13:45 ---------------
```

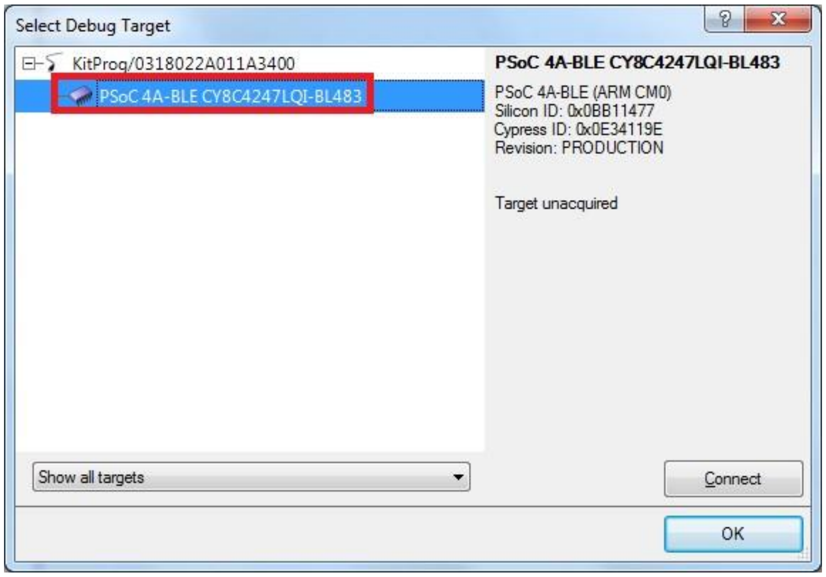
3. Select **Debug** > **Select Debug Target**, as shown in Figure 7.

Figure 7. Selecting Debug Target



4. In the **Select Debug Target** dialog box, click **Port Acquire**, and then click **Connect** as shown in Figure 8. Click **OK** to close the dialog box.

Figure 8. Connecting to a Device



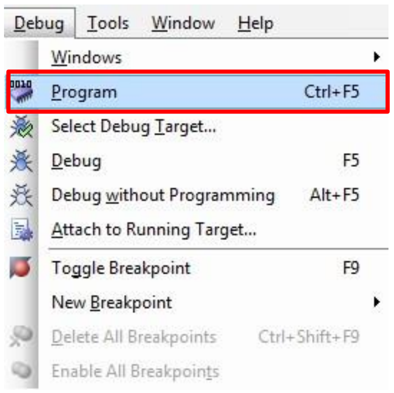
PSoC 4A-BLE CY8C4247LQI-BL483
PSoC 4A-BLE (ARM CM0)
Silicon ID: 0x0BB11477
Cypress ID: 0x0E34119E
Revision: PRODUCTION

Target unacquired

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.

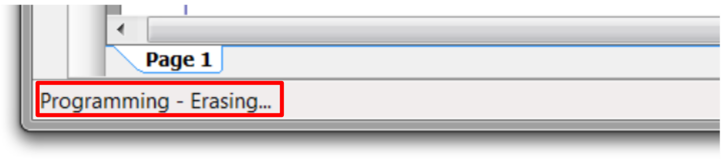
5. Select **Debug** > **Program** to program the device with the project, as shown in Figure 9.

Figure 9. 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 10.

Figure 10. Programming Status

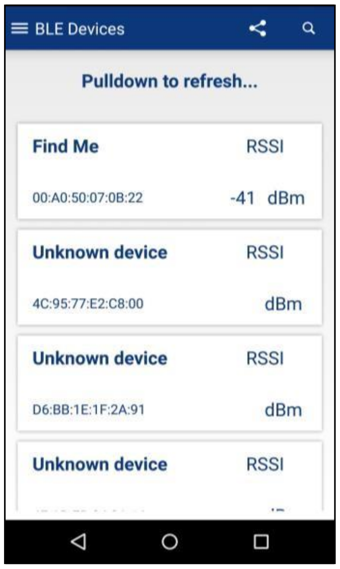


Testing

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

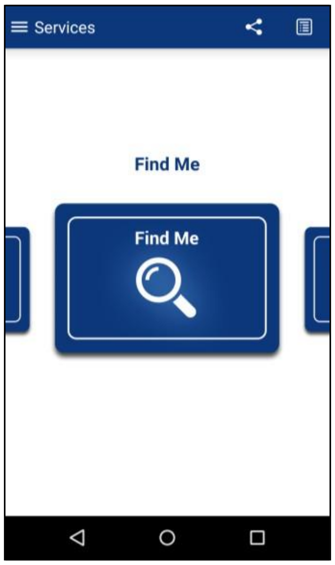
1. Plug the BLE-USB Bridge (included with the BLE Pioneer Kit) in your computer's USB port.
2. On your BLE-enabled mobile phone, open the **CySmart app** (available on the iOS and Android app stores)
3. Once the app is open, **swipe down** to refresh the list of nearby advertising BLE devices. See Figure 11.

Figure 11: CySmart App Scanning for BLE Devices



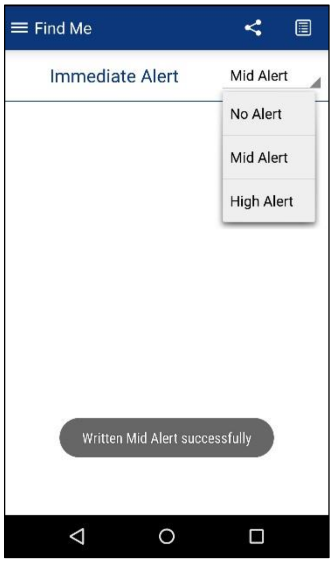
4. **Tap** on the **Find Me** device to connect to it.
5. **Swipe right** to see the **Find Me Profile** and tap on it to open the Immediate Alert Service. See Figure 12.

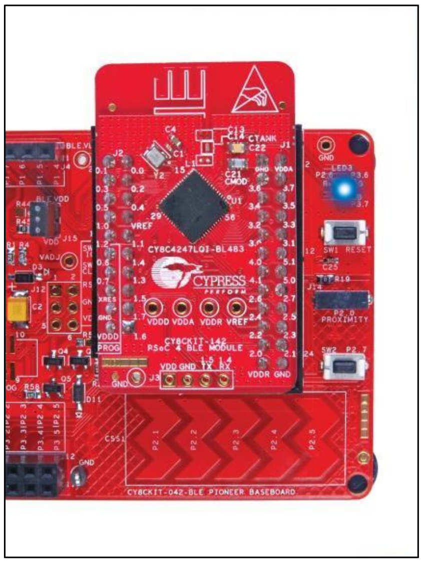
Figure 12: CySmart App Find Me Service Tab



6. **Select** from the **No Alert**, **Mid Alert**, or **High Alert** options to change the LED mode on the BLE Pioneer Kit. See Figure 13.

Figure 13: LED on BLE Pioneer Kit Controlled via Immediate Alert Level in CySmart App

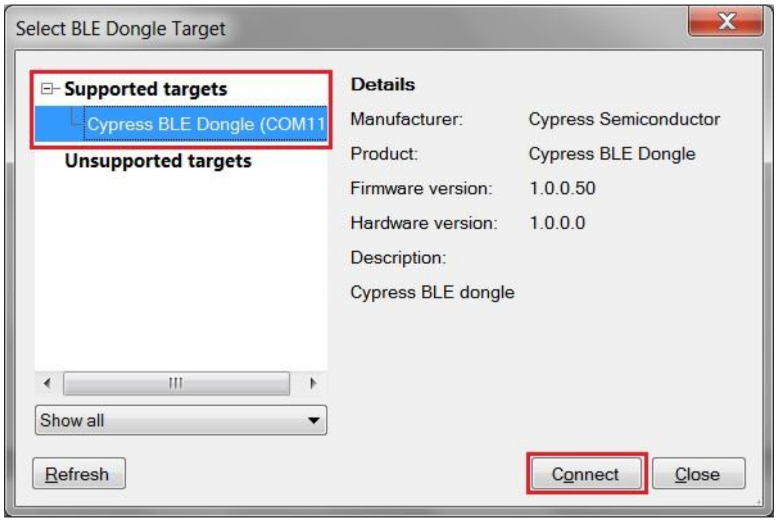




Testing with the CySmart BLE Test and Debug Utility for Windows PC:

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 14.

Figure 14: CySmart: Select BLE Dongle Target



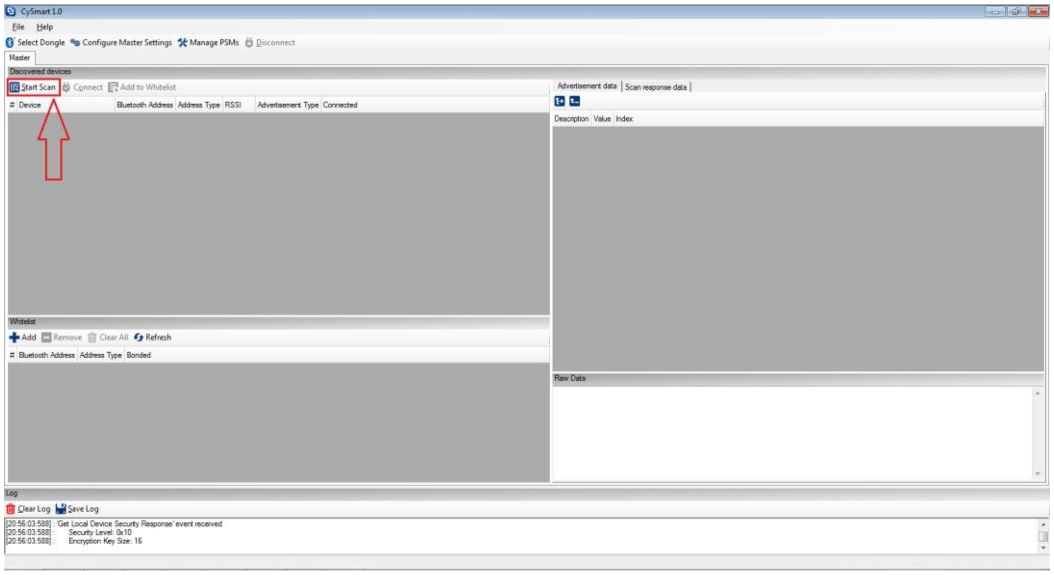
Details

Manufacturer:	Cypress Semiconductor
Product:	Cypress BLE Dongle
Firmware version:	1.0.0.50
Hardware version:	1.0.0.0
Description:	

Cypress BLE dongle

3. When the BLE-USB Bridge is connected, click on **Start Scan** to find your BLE device. See Figure 15.

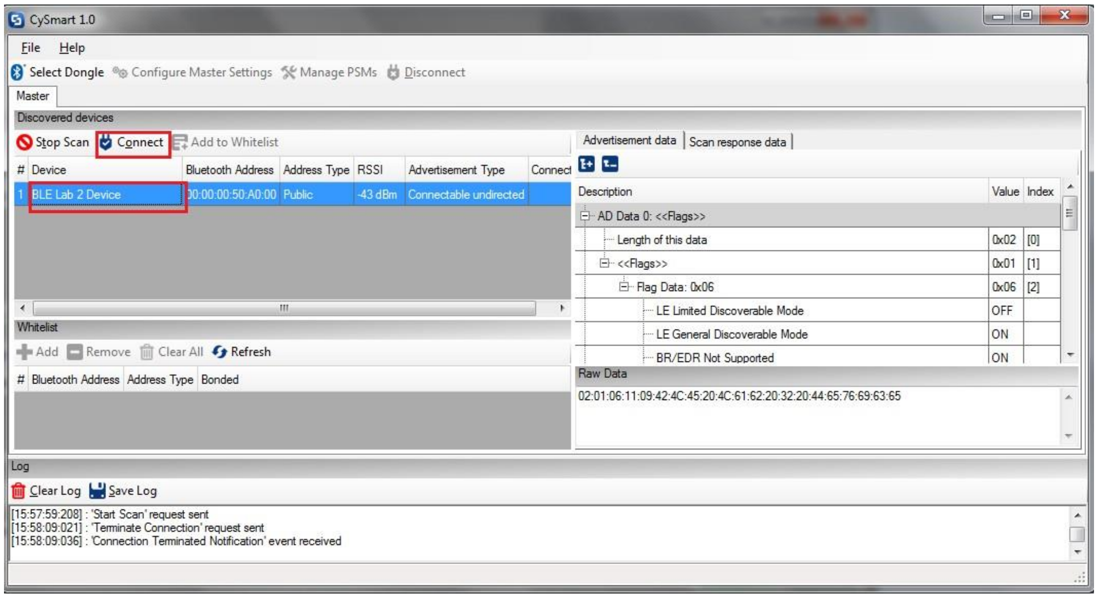
Figure 15: Finding a BLE Device


CySmart 1.0
File Help
Select Dongle Configure Master Settings Manage PSMs Disconnect
Master
Discovered devices
Start Scan Connect Add to Whitelist
# Device Bluetooth Address Address Type RSSI Advertisement Type Connected
Advertisement data Scan response data
Description Value Index
Whitelist
Add Remove Clear All Refresh
# Bluetooth Address Address Type Bonded
Raw Data
Log
Clear Log Save Log

```
[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 scanning stops automatically once all the nearby devices are known. The tool lists all the nearby devices in the Discovered devices section.
5. Click on your device name to see the Advertisement data and Scan response data packets on the right. See Figure 16.

Figure 16: Checking Discovery Details of a Connected BLE 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
# Bluetooth Address Address Type Bonded
Log
Clear Log Save Log

#	Device	Bluetooth Address	Address Type	RSSI	Advertisement Type	Connect
1	BLE Lab 2 Device	00:00:00:50:A0:00	Public	-43 dBm	Connectable undirected	

Description	Value	Index
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	

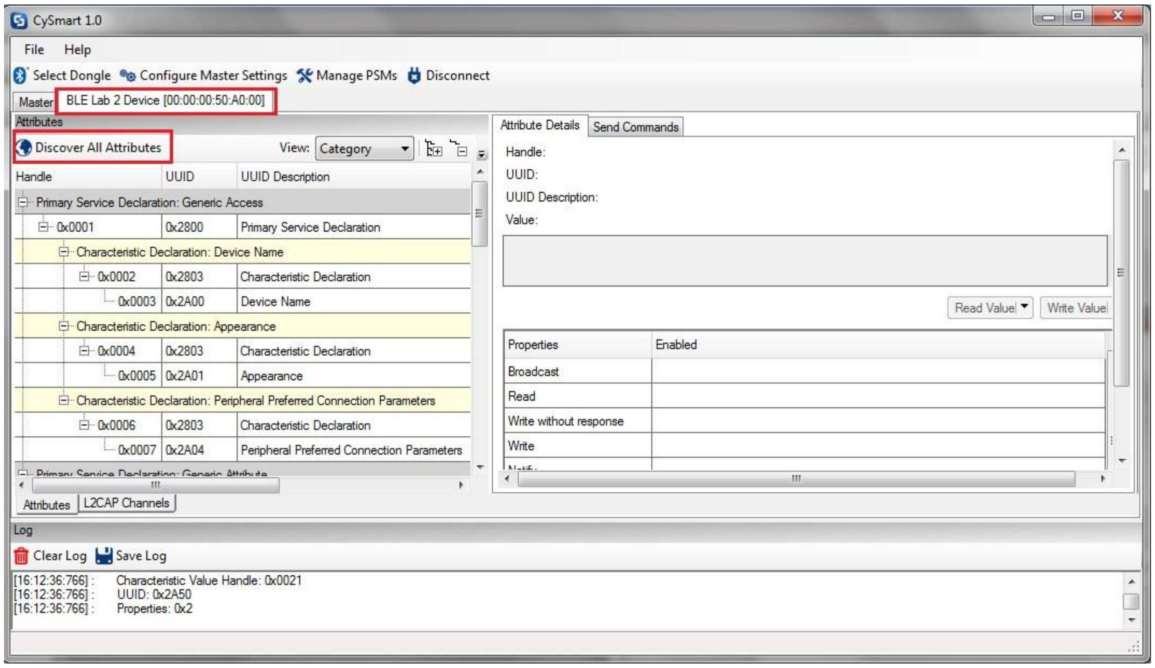
Raw Data

```
02:01:06:11:09:42:4C:45:20:4C:61:62:20:32:20:44:65:76:69:63:65
```

```
[15:57:59:208] : 'Start Scan' request sent
[15:58:09:021] : 'Terminate Connection' request sent
[15:58:09:036] : 'Connection Terminated Notification' event received
```

6. Click **Connect** as seen in Figure 16 to connect to the device.
7. The tool will now open a separate tab for the device. Click **Discover All Attributes** to list all the Attributes in the device, with their respective UUIDs and descriptions. See Figure 17.

Figure 17: Discovering Attributes of a Connected BLE Device



Master | BLE Lab 2 Device [00:00:00:50:A0:00]

Attributes

Handle	UUID	UUID Description
Primary Service Declaration: Generic Access		
0x0001	0x2800	Primary Service Declaration
Characteristic Declaration: Device Name		
0x0002	0x2803	Characteristic Declaration
0x0003	0x2A00	Device Name
Characteristic Declaration: Appearance		
0x0004	0x2803	Characteristic Declaration
0x0005	0x2A01	Appearance
Characteristic Declaration: Peripheral Preferred Connection Parameters		
0x0006	0x2803	Characteristic Declaration
0x0007	0x2A04	Peripheral Preferred Connection Parameters

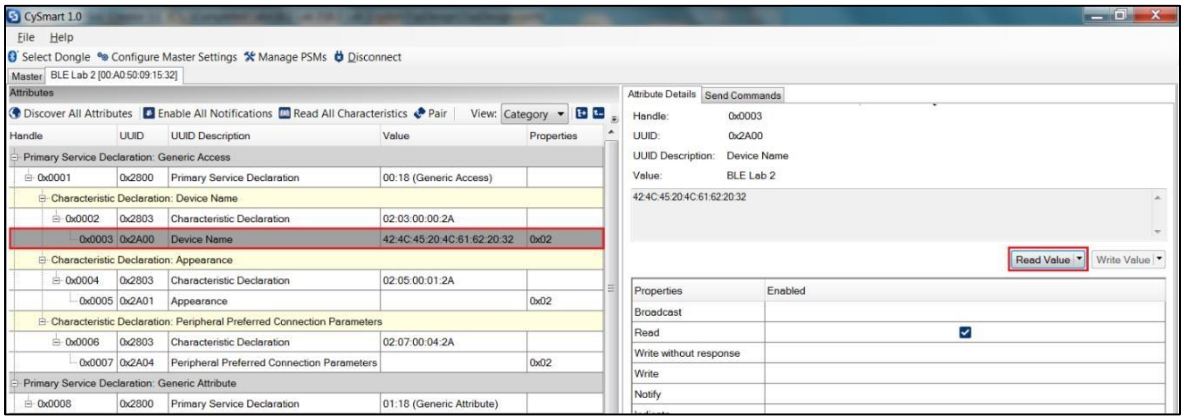
Handle:
UUID:
UUID Description:
Value:

Properties	Enabled
Broadcast	
Read	
Write without response	
Write	

```
[16:12:36:766] :    Characteristic Value Handle: 0x0021
[16:12:36:766] :    UUID: 0x2A50
[16:12:36:766] :    Properties: 0x2
```

8. Click on any row in the list of Attributes to see its details on the right. To read an Attribute's value, click **Read Value** on the right as shown in Figure 18.

Figure 18: Reading Attribute Value



Master | BLE Lab 2 [00:A0:50:09:15:32]

Handle	UUID	UUID Description	Value	Properties
Primary Service Declaration: Generic Access				
0x0001	0x2800	Primary Service Declaration	00:18 (Generic Access)	
Characteristic Declaration: Device Name				
0x0002	0x2803	Characteristic Declaration	02:03:00:00:2A	
0x0003	0x2A00	Device Name	42:4C:45:20:4C:61:62:20:32	0x02
Characteristic Declaration: Appearance				
0x0004	0x2803	Characteristic Declaration	02:05:00:01:2A	
0x0005	0x2A01	Appearance		0x02
Characteristic Declaration: Peripheral Preferred Connection Parameters				
0x0006	0x2803	Characteristic Declaration	02:07:00:04:2A	
0x0007	0x2A04	Peripheral Preferred Connection Parameters		0x02
Primary Service Declaration: Generic Attribute				
0x0008	0x2800	Primary Service Declaration	01:18 (Generic Attribute)	

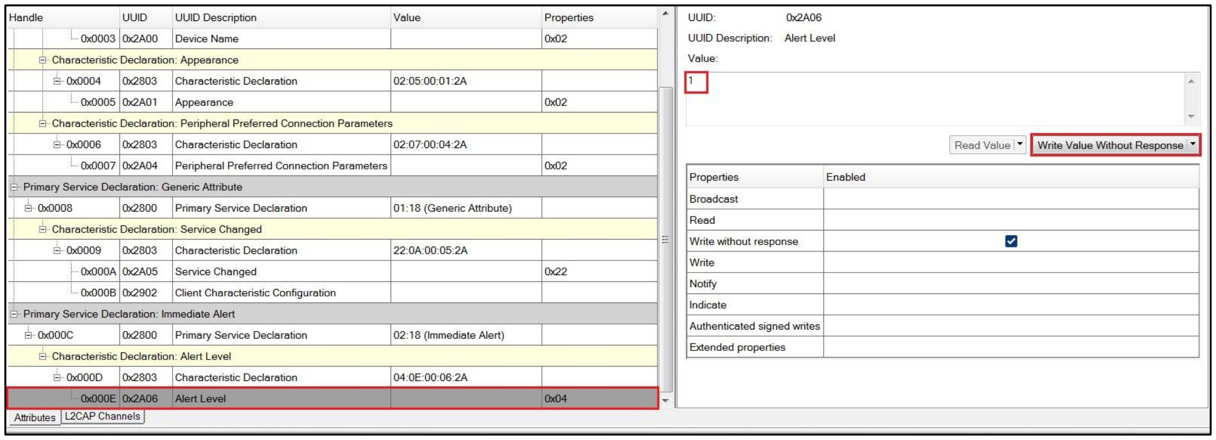
Handle: 0x0003
UUID: 0x2A00
UUID Description: Device Name
Value: BLE Lab 2

42:4C:45:20:4C:61:62:20:32

Properties	Enabled
Broadcast	
Read	✓
Write without response	
Write	
Notify	

9. Locate the **Alert Level** Attribute for the **Immediate Alert Service**. On the right, write a value of **1** to start blinking the LED. See Figure 19.

Figure 19: Writing Attribute Value



Handle	UUID	UUID Description	Value	Properties
0x0003	0x2A00	Device Name		0x02
Characteristic Declaration: Appearance				
0x0004	0x2803	Characteristic Declaration	02:05:00:01:2A	
0x0005	0x2A01	Appearance		0x02
Characteristic Declaration: Peripheral Preferred Connection Parameters				
0x0006	0x2803	Characteristic Declaration	02:07:00:04:2A	
0x0007	0x2A04	Peripheral Preferred Connection Parameters		0x02
Primary Service Declaration: Generic Attribute				
0x0008	0x2800	Primary Service Declaration	01:18 (Generic Attribute)	
Characteristic Declaration: Service Changed				
0x0009	0x2803	Characteristic Declaration	22:0A:00:05:2A	
0x000A	0x2A05	Service Changed		0x22
0x000B	0x2902	Client Characteristic Configuration		
Primary Service Declaration: Immediate Alert				
0x000C	0x2800	Primary Service Declaration	02:18 (Immediate Alert)	
Characteristic Declaration: Alert Level				
0x000D	0x2803	Characteristic Declaration	04:0E:00:06:2A	
0x000E	0x2A06	Alert Level		0x04

UUID: 0x2A06
UUID Description: Alert Level
Value:
1

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

10. Write a value of **2** to keep the LED always on.
11. Write a value of **0** to turn off the LED.

Related Documents

Table 2 lists all relevant application notes, code examples, knowledge base articles, device datasheets, and Component / user module datasheets.

Table 2. 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.