

ECE9407– PSoC 4 BLE Lab #2

BLE Connection

Name: Amr Gaballah, An He
Number: 250776065, 250847871
Data: 2016-06-06
Instructor: Dr. Ahmed Hussein

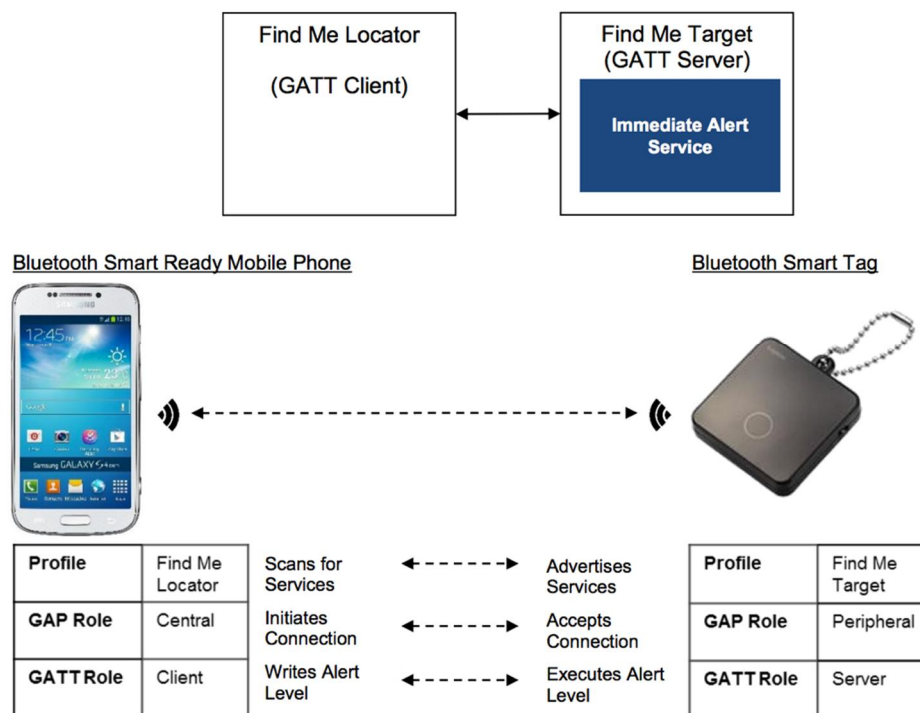
1. Lab1 Task Description

This lab aims to give an introduction to setting up a BLE connection using PSOC Creator 3.1. The lab aims to create a localization scenario using the find me profile installed on the PSOC creator. BLE Find Me profile defines how pressing a button on one BLE device causes an alerting signal on another BLE device. This can be used to find misplaced devices.

The two BLE roles defined by find me profile are:

1. Find Me Locator (GATT Client): is the device initiating the alert signal.
2. Find Me Server (GATT Server): is the device that receives the alerting message and triggers a user alert. This is shown in the graph in LAB 2 manual

Figure 1: BLE Find Me Profile Roles



The BLE Pioneer Kit will act as the Find Me Target

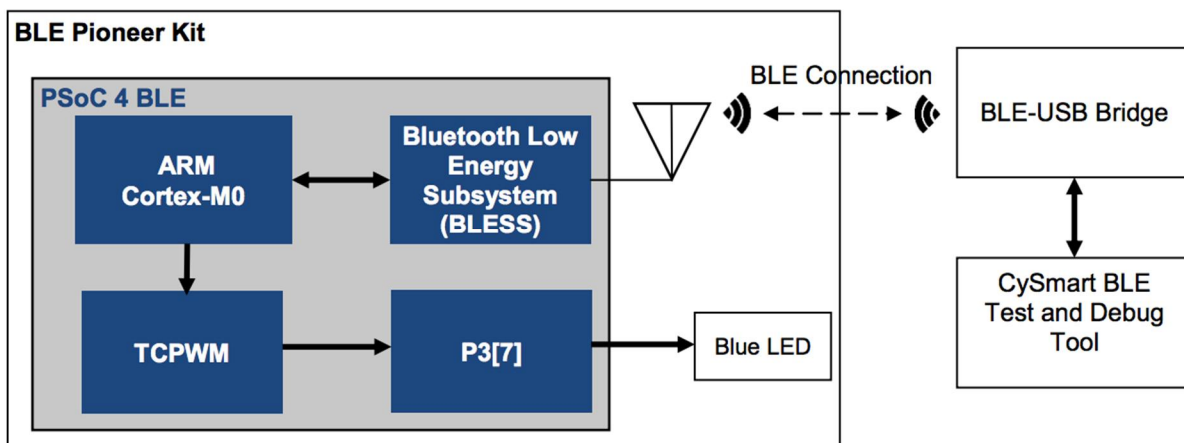
2. Equipment and Objectives



While the BLE-USB Bridge will act the Find Me Server

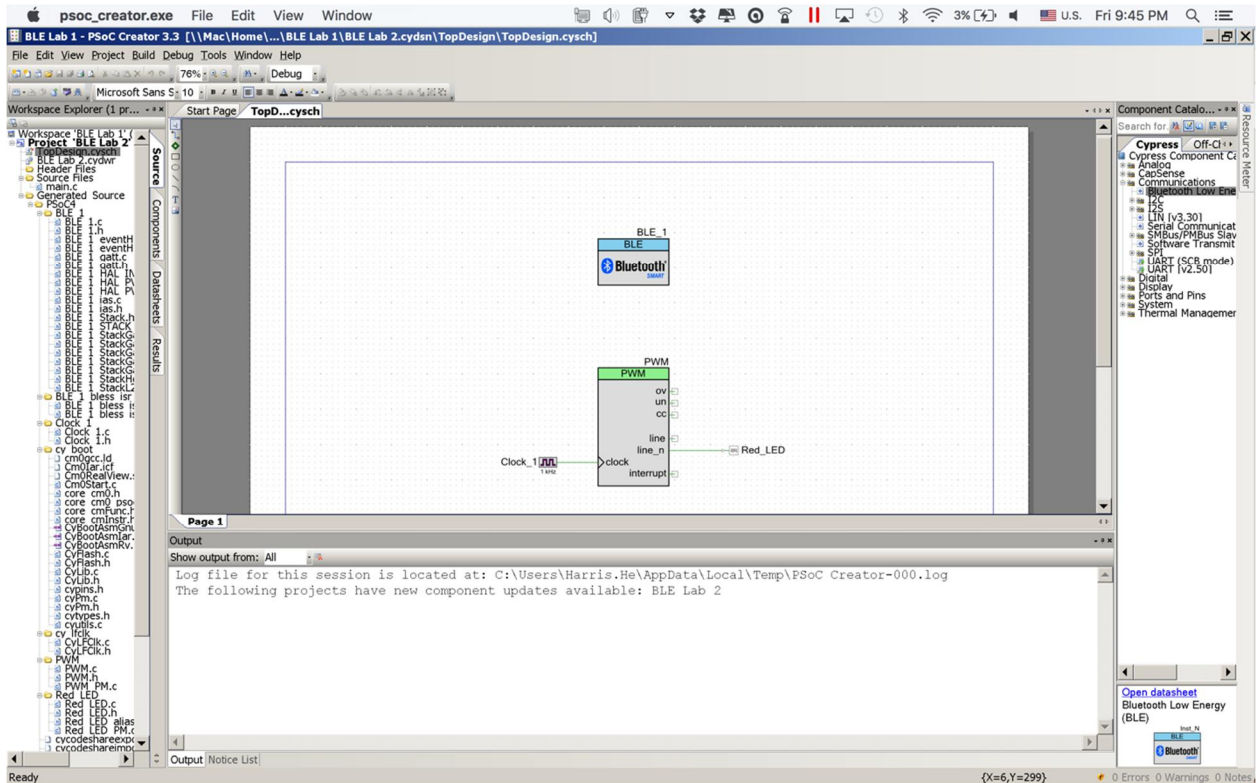


Below is the block diagram of LAB 2 project from LAB 2 manual

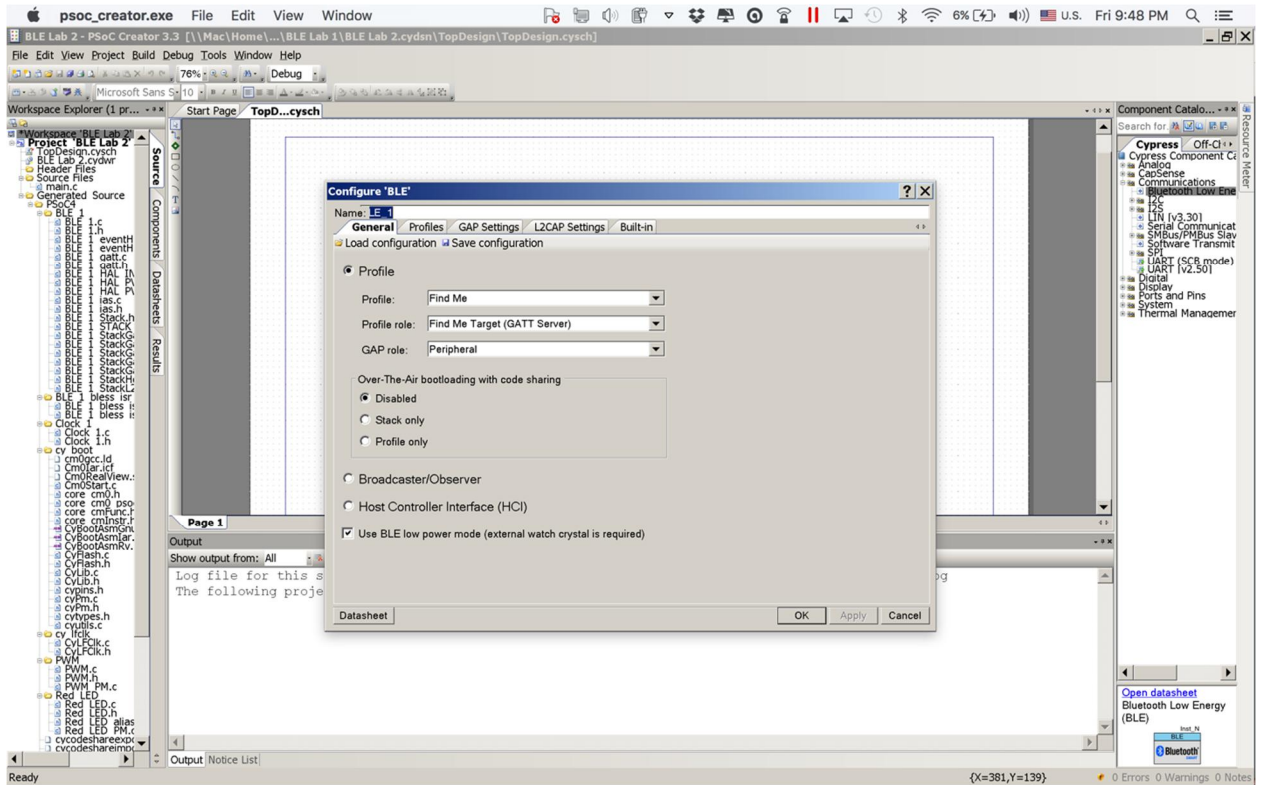


3. Steps and Results

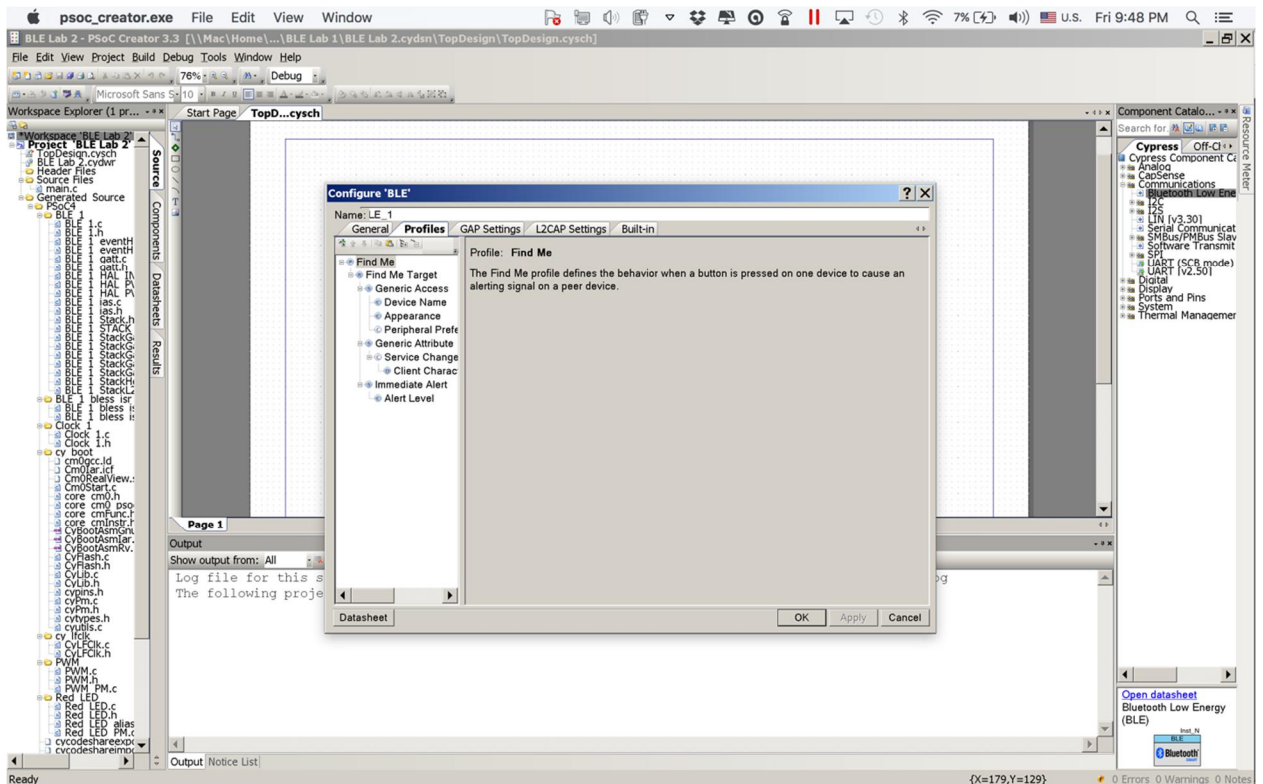
3.1 Drag the BLE component to TopDesign.cysch from the Workspace Explorer as in the screenshot below:



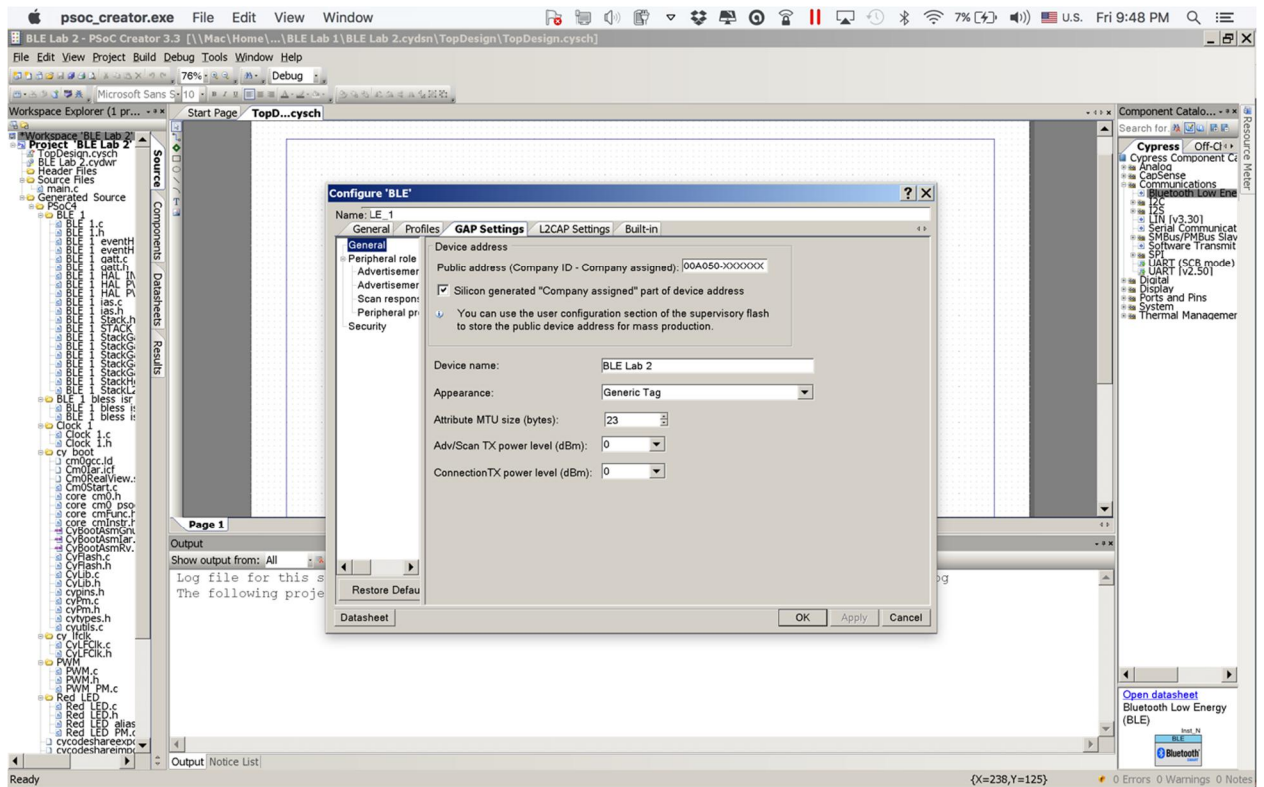
3.2 Click on the BLE component to open the configuration. Change the profile to Find Me, and make the role a Find Me Target. This is shown in the figure below:



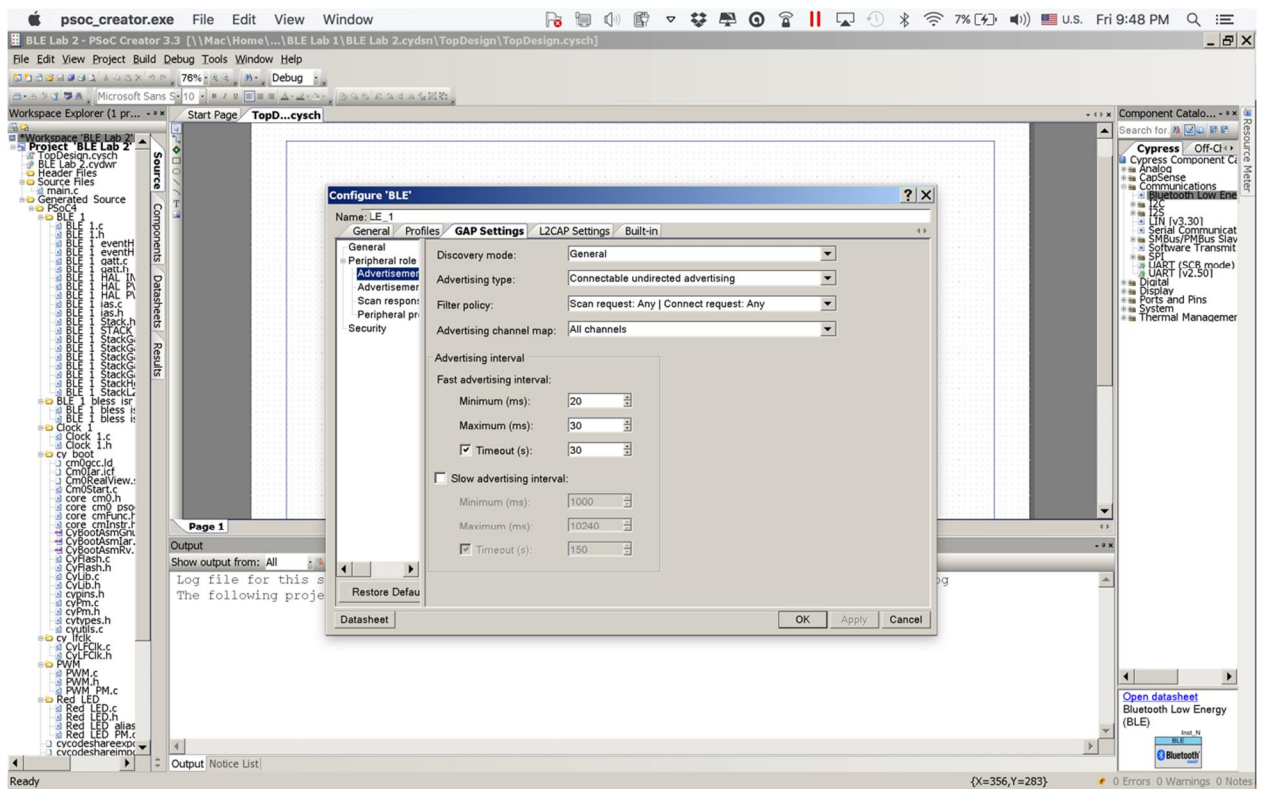
3.3 No changes are done in the profile tab as below:



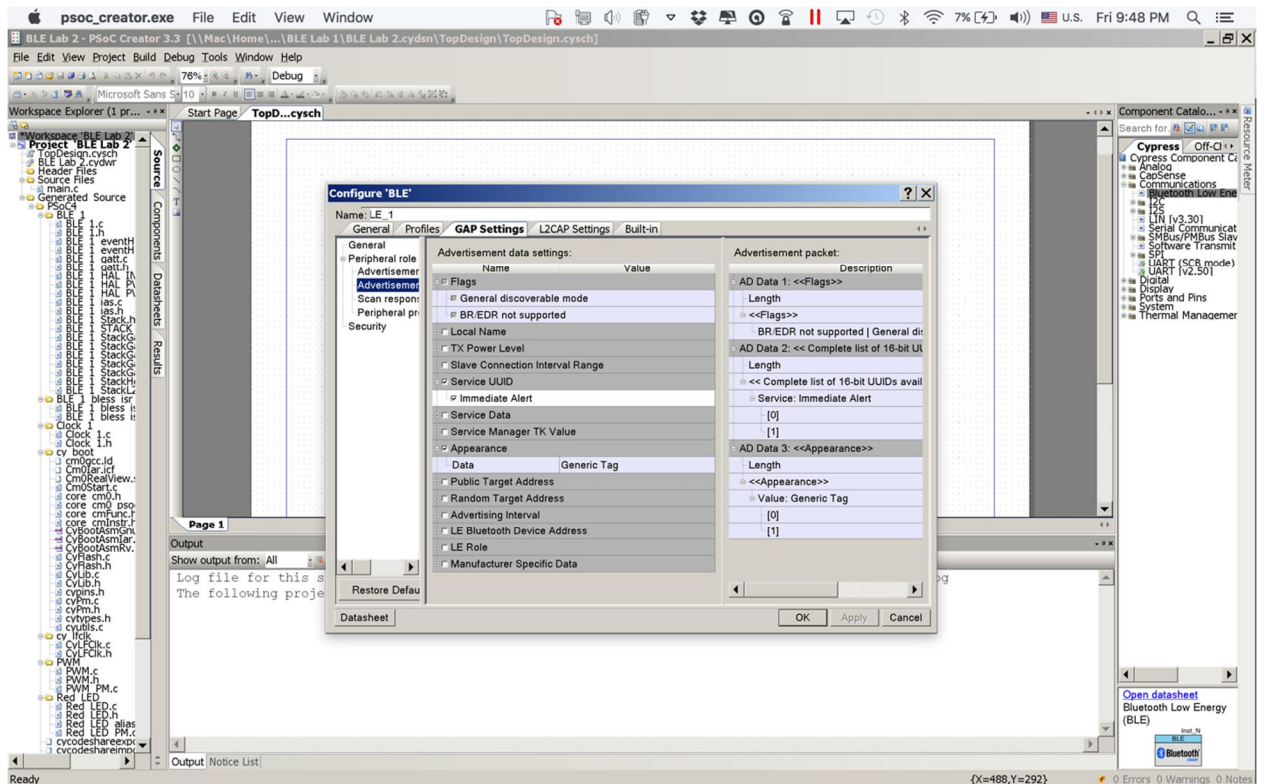
3.4 Click on the GAP settings, and change the device name and appearance as below:



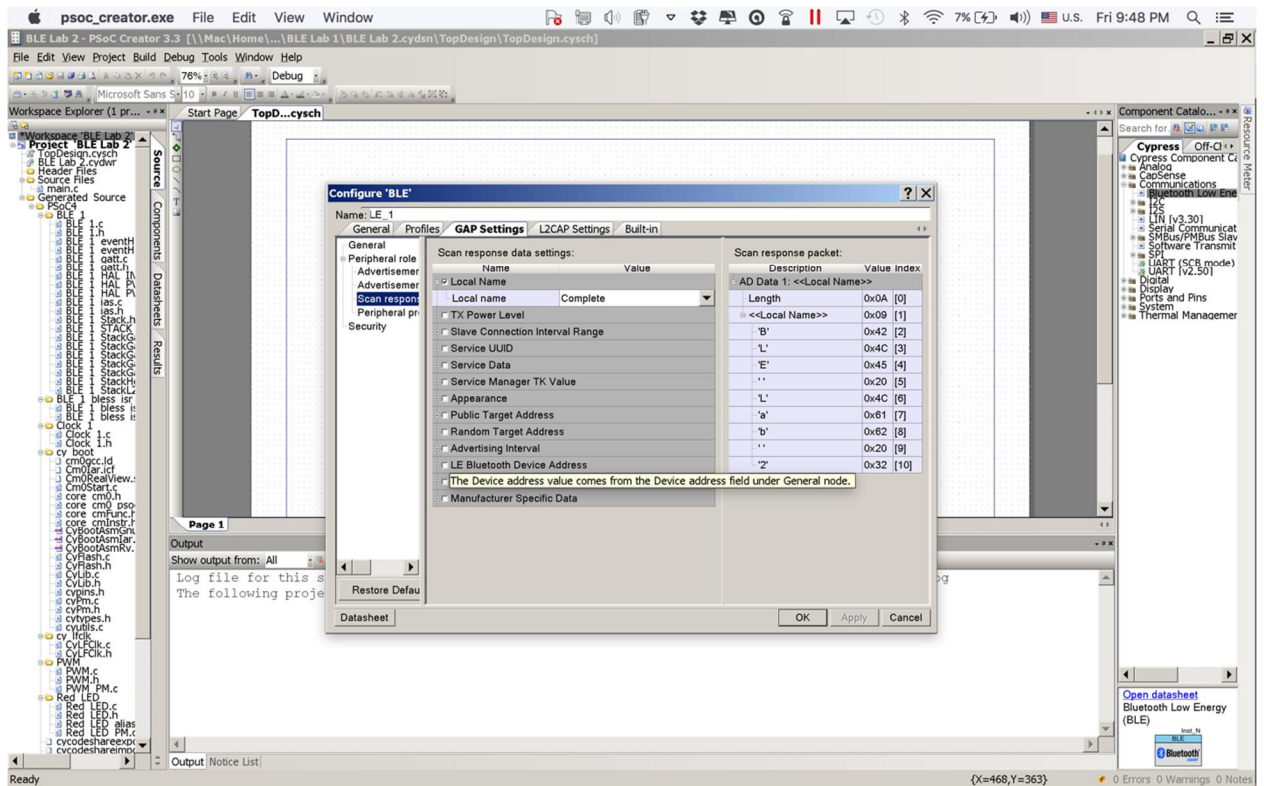
3.5 From the left click on advertisement settings then change the settings as below:



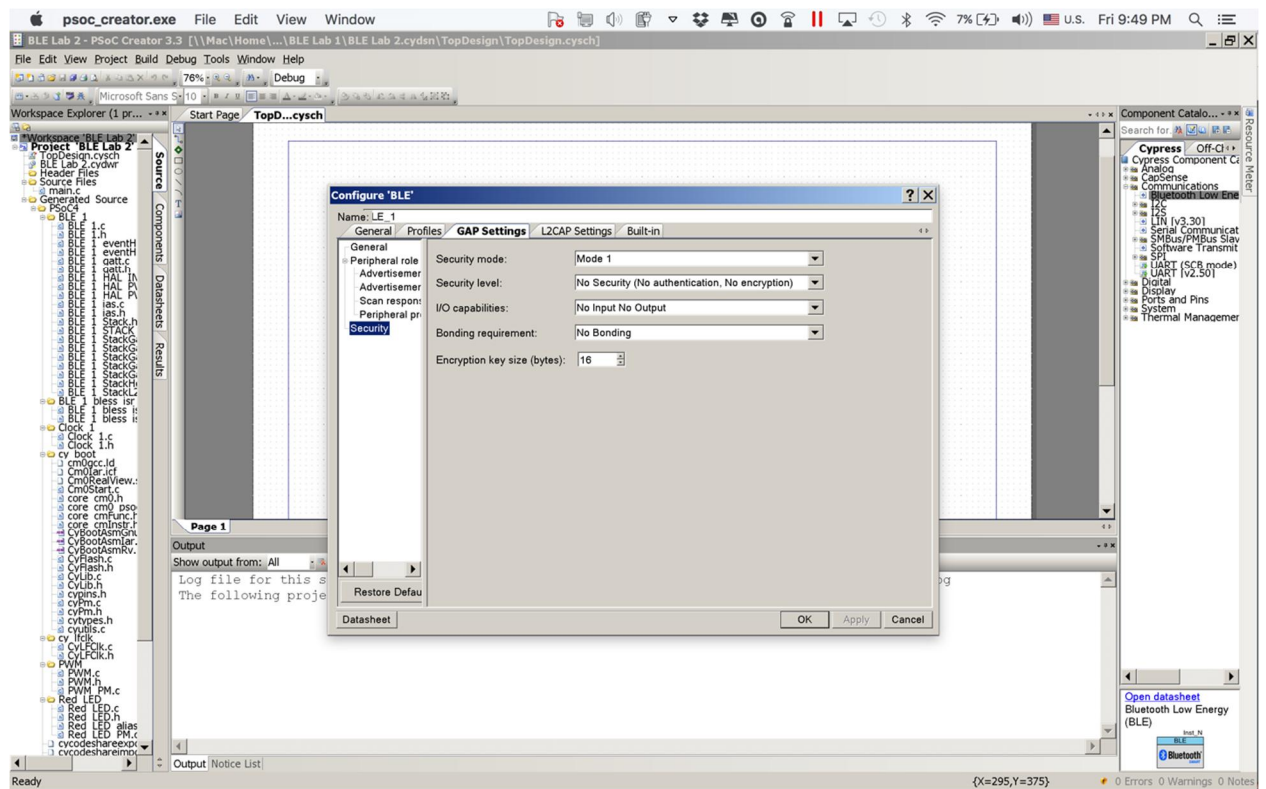
3.6 Click on Advertisement Packet from the left then apply the changes below:



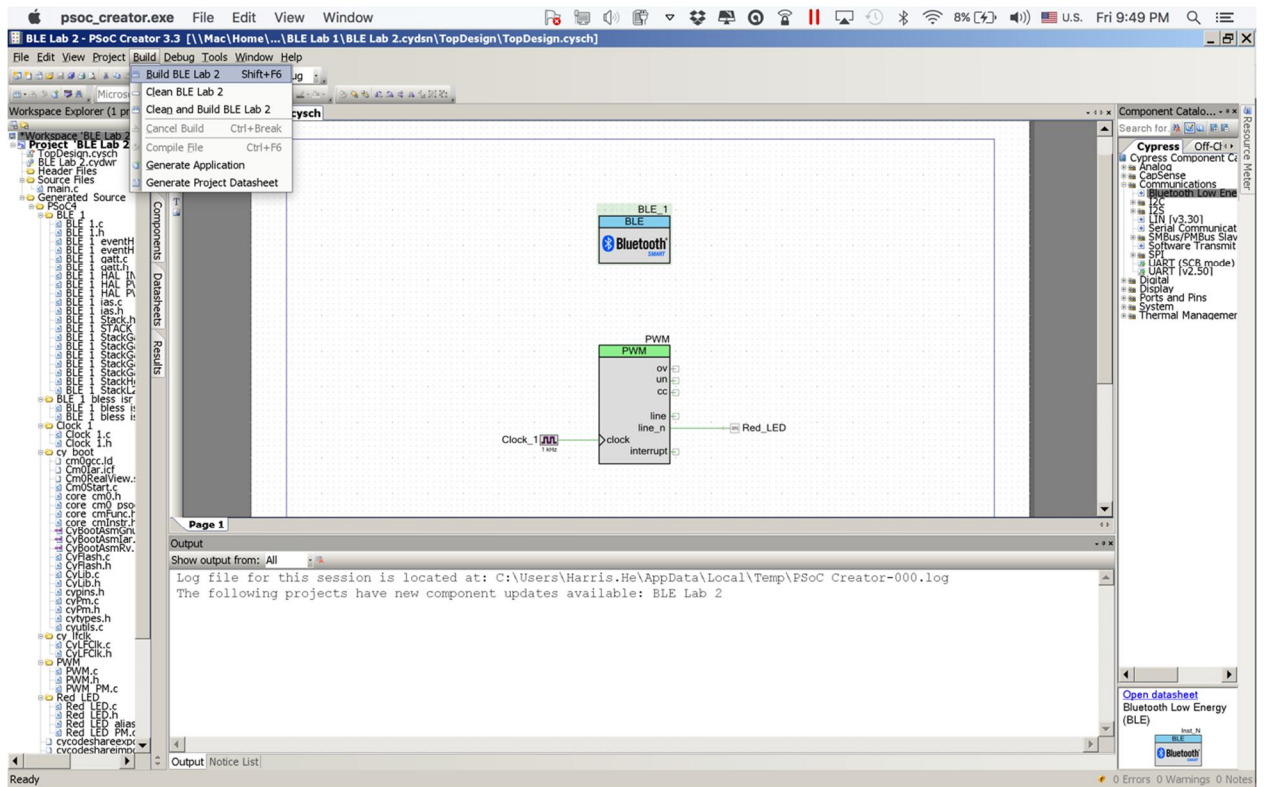
3.7 Change the properties of the scan response as below:



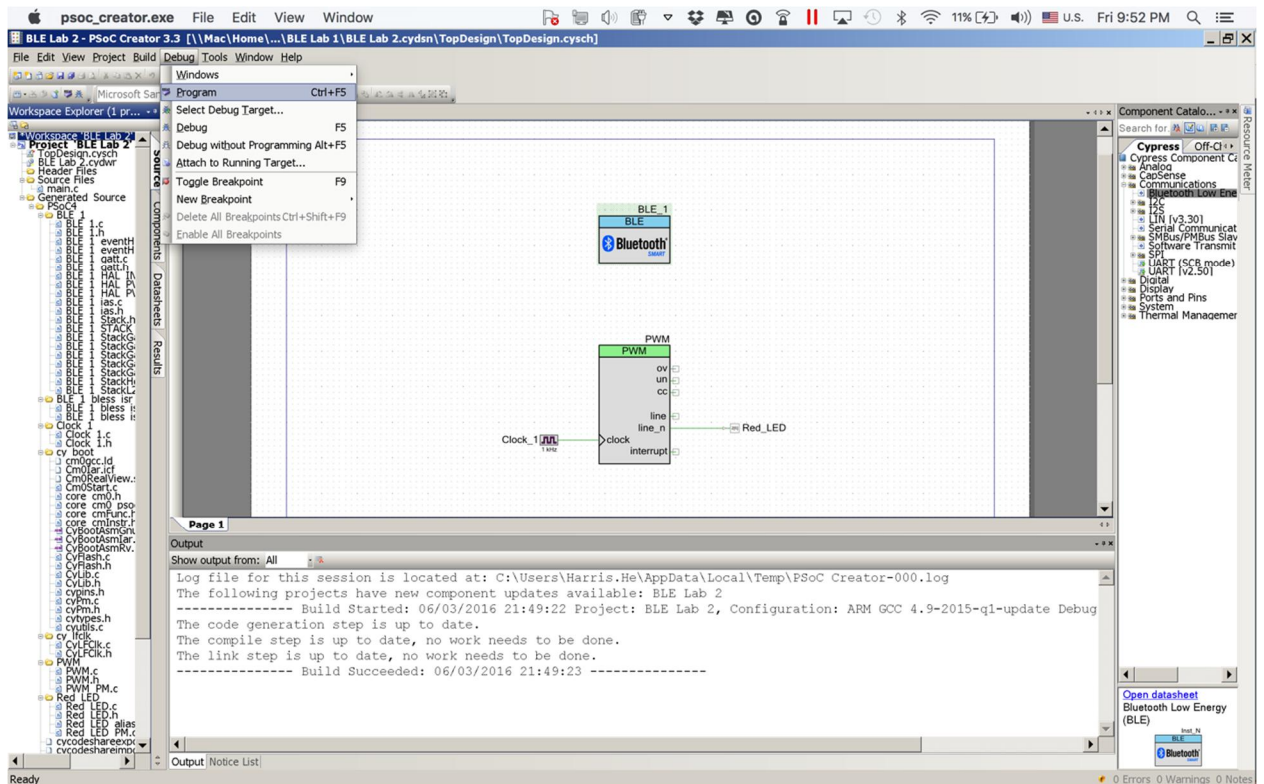
3.8 Change the security settings as below, then click ok to save and close the configuration settings.



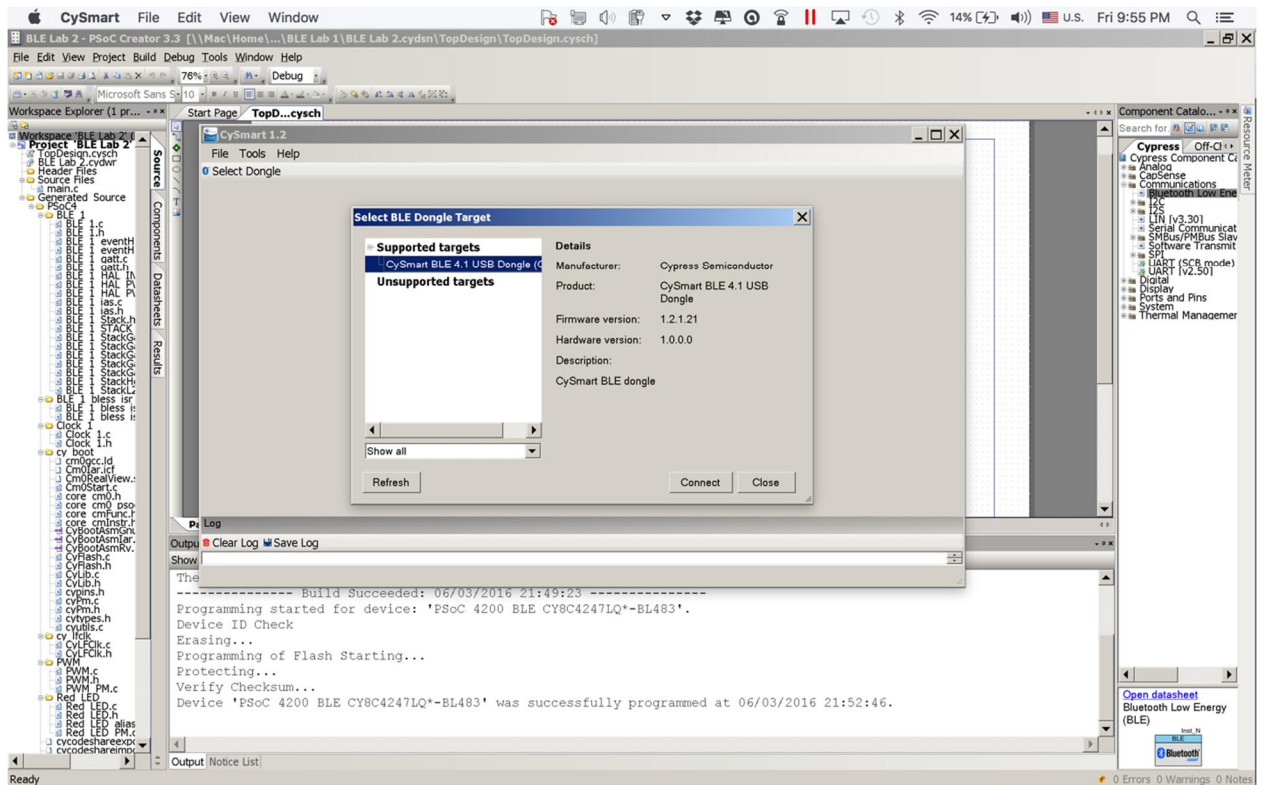
3.9 Click on Build the select Build BLE Lab 2 to compile the C code, make sure there are no errors after compiling the code.



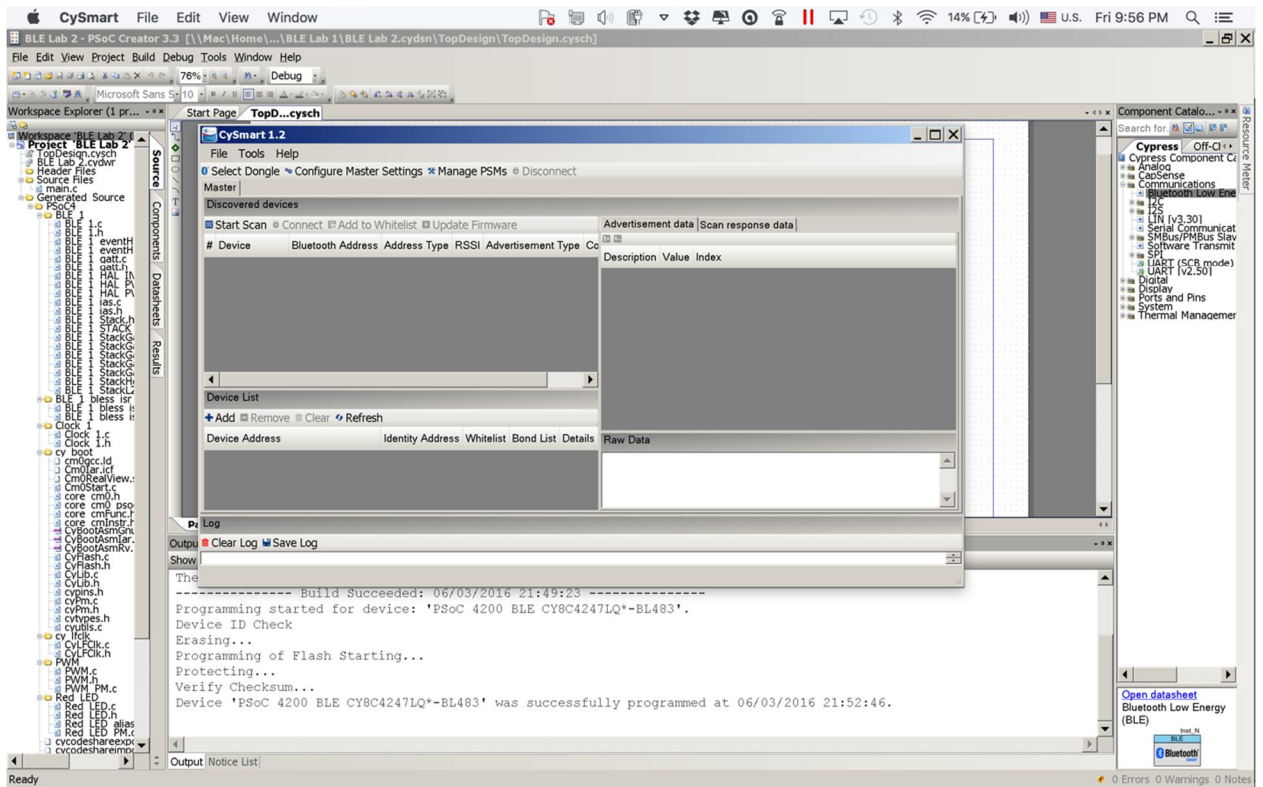
3.10 Click on Debug, then program to program the BLE Pioneer Kit with the code



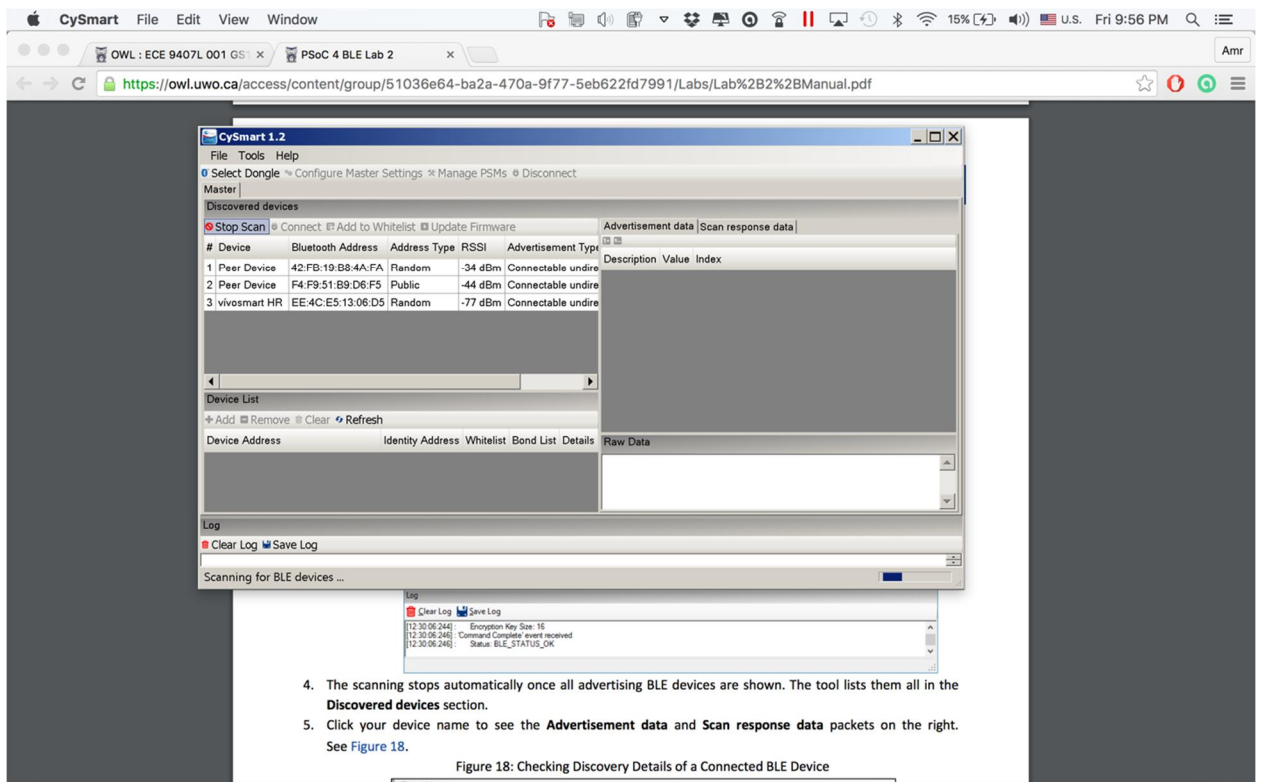
3.11 Insert the BLE-USB bridge into the other USB port and then open CYSMART 1.2 tool, and select the BLE-USB bridge device as below:



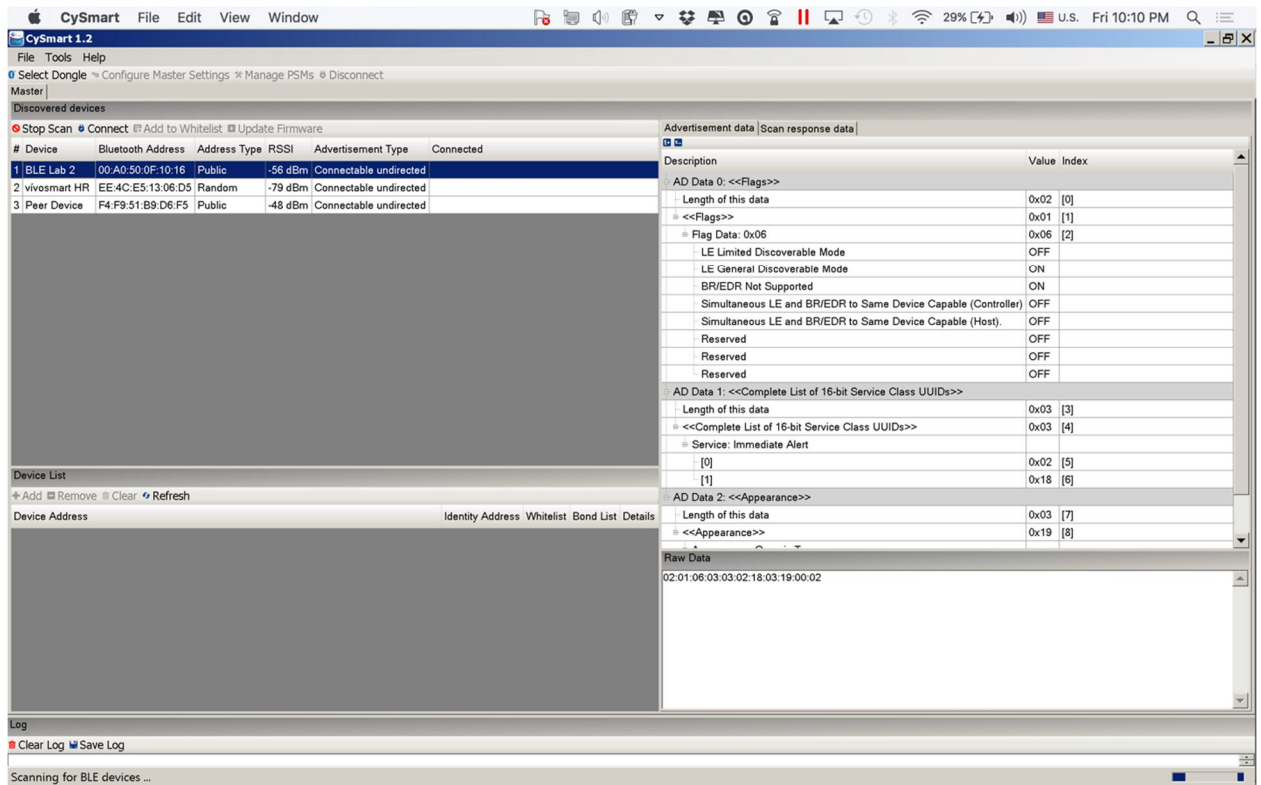
3.12 The following figure will appear, click start scan to search for the BLE Pioneer Kit.



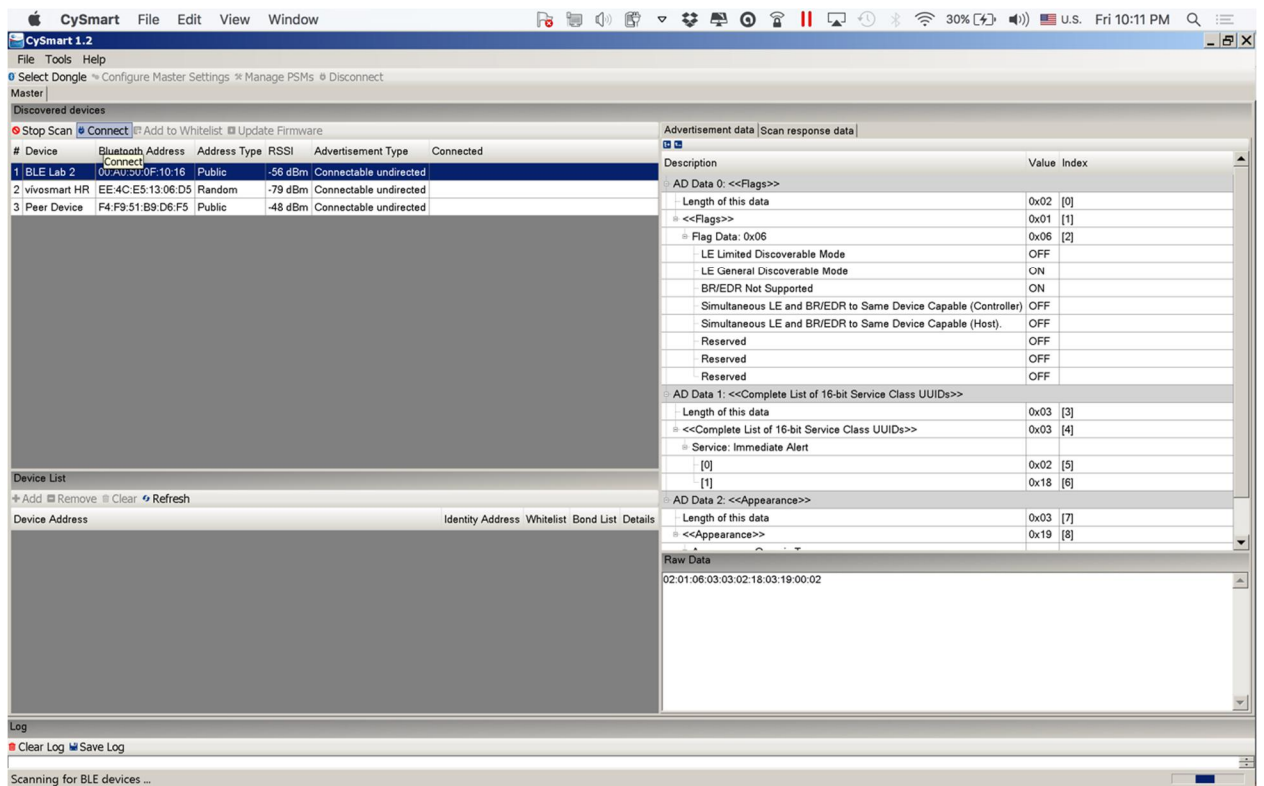
3.13 The program will continue to scan until it finds the other BLE device as below:



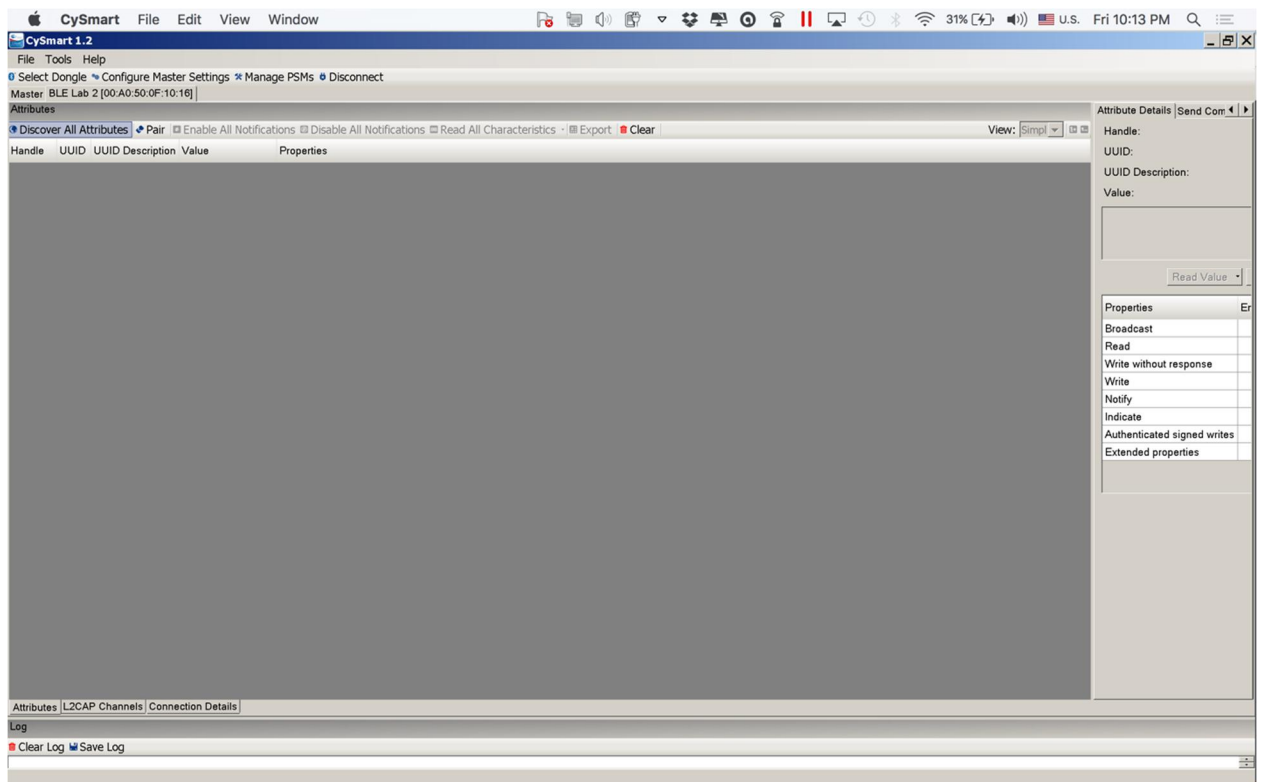
3.14 Click on the BLE Pioneer Kit as below to check exchanged messages.



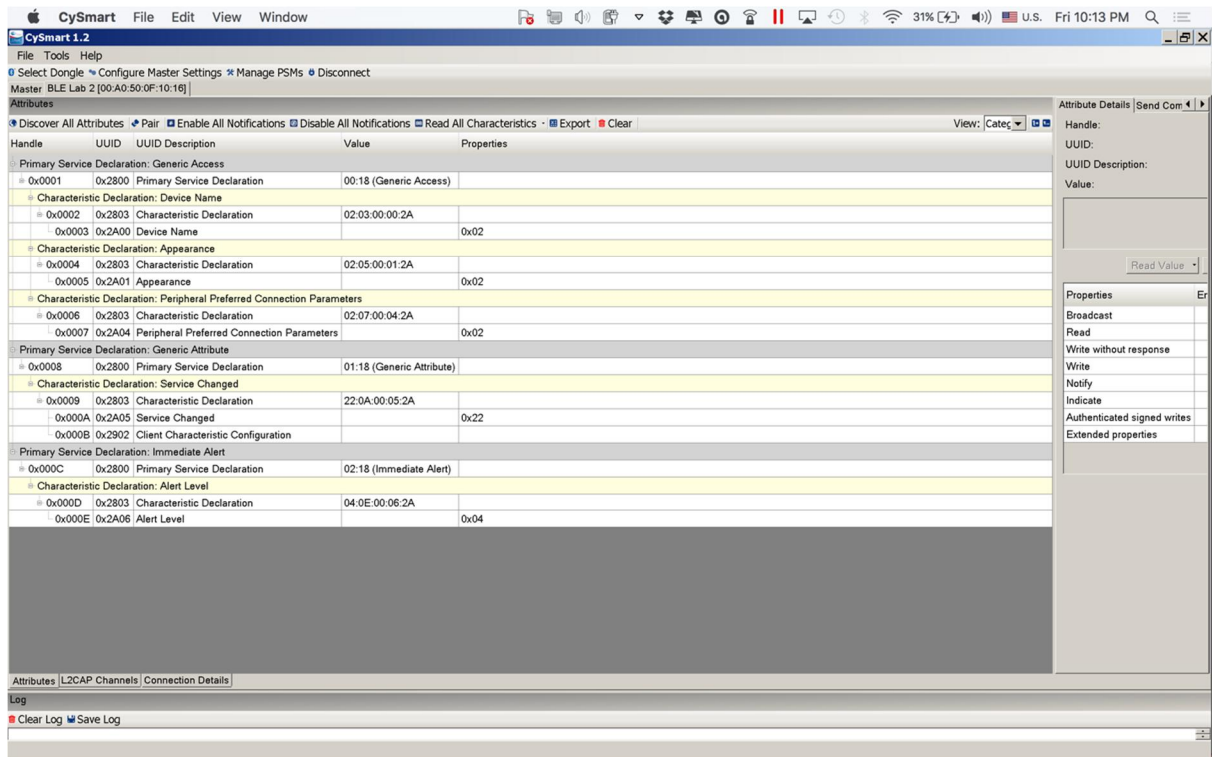
3.15 Click connect to connect BLE-USB bridge to the BLE Pioneer Kit.



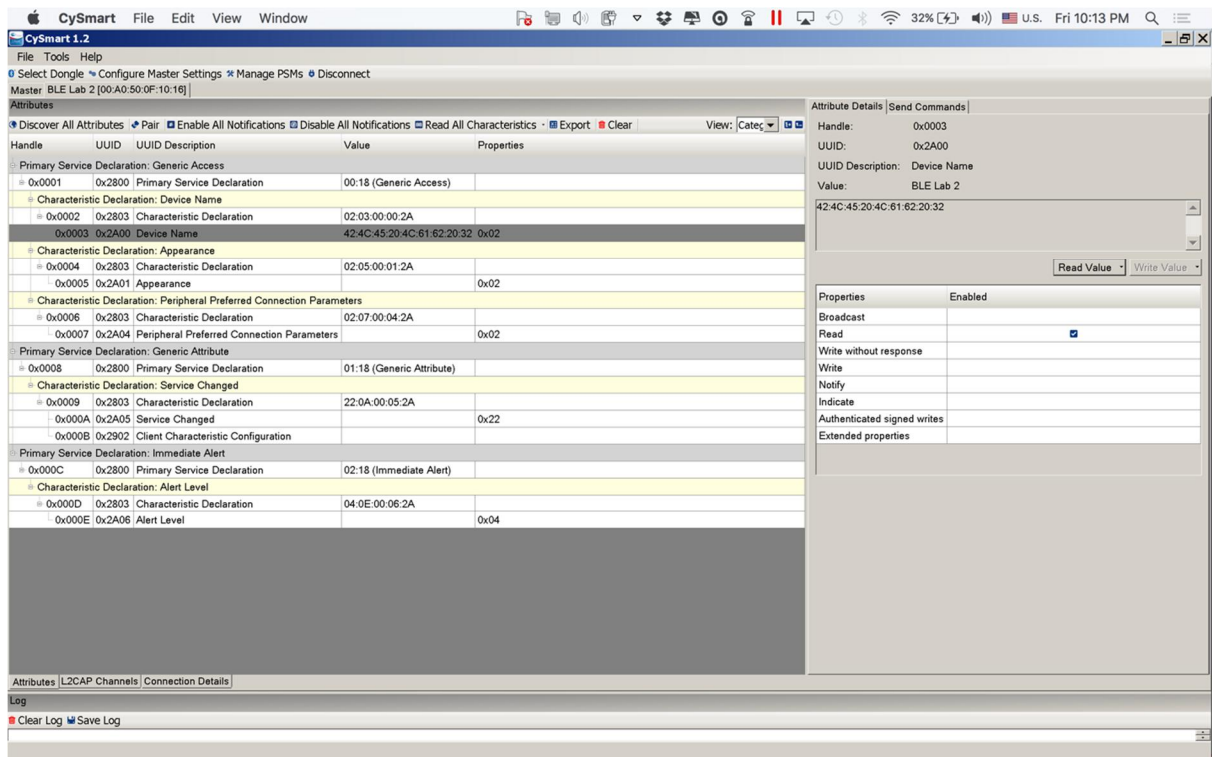
3.16 After connection, click on discover all attributes to check the messages



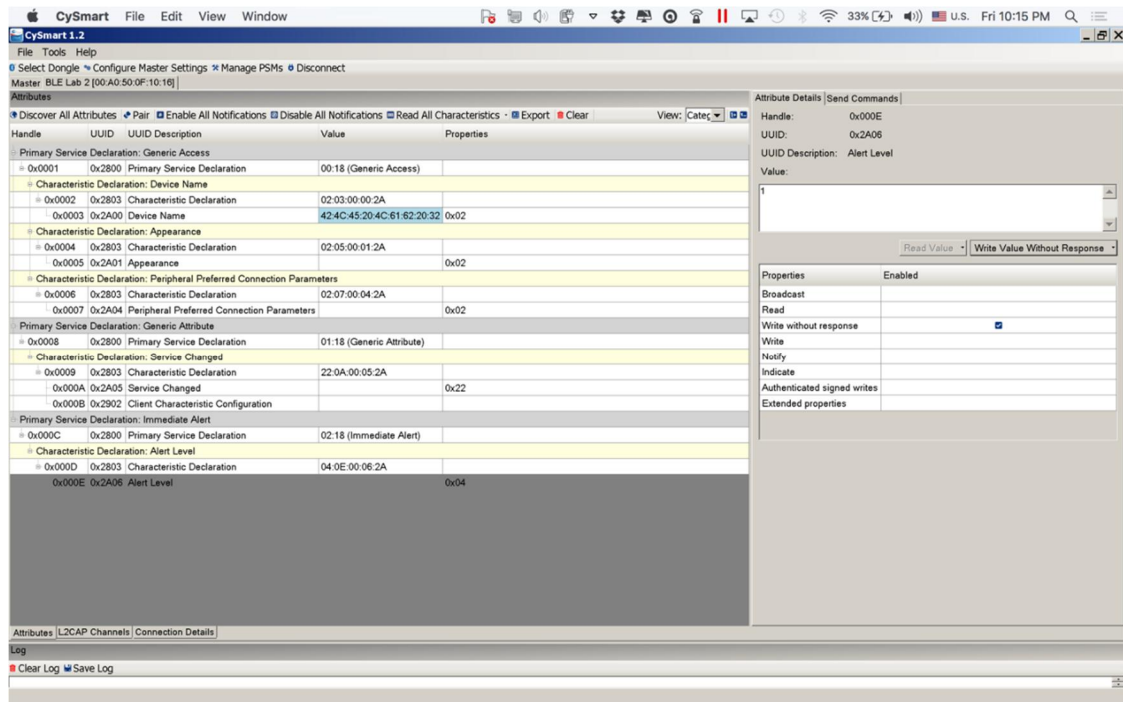
3.17 The attributes will appear as below



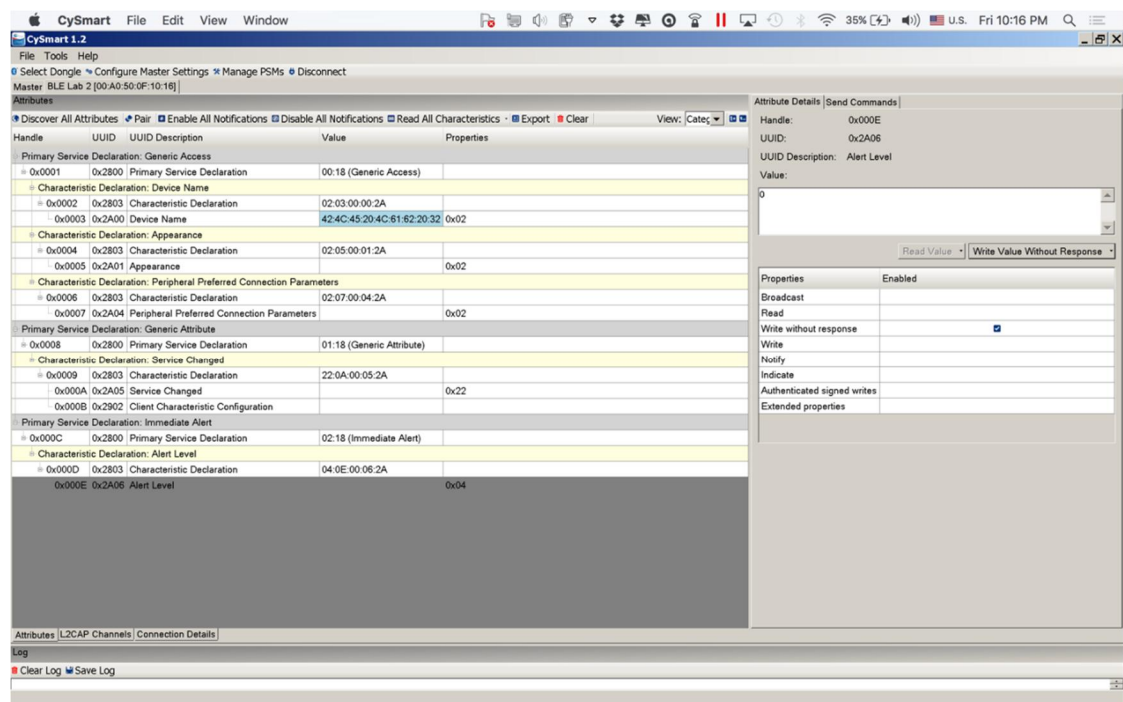
3.18 From the left, click on the device name tab, then click on read value to check the device name



3.19 To change the alert level, click on alert level from the left, then click write value from the right to change the value

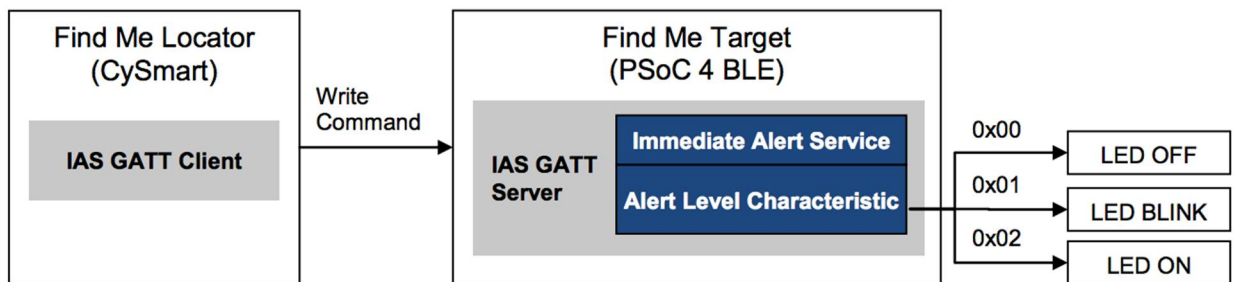


3.20 The value 1 indicates the blinking alert LED, the value 2 indicates the constant alert LED, while the value 0 indicates the no alert case.



4. Conclusion:

The BLE Kit acted as the Find Me server. It was detected by CYSMART tool, and the connection was established. The Find Me server contained the alarm levels and their meaning as a part of the C code. When we changed the alarm level in CYSMART tool, there was a corresponding change on the kit as the following diagram from LAB 2 manual explains.



We also study and research why BLE can control LED through PWM. The reason is that the firmware (main.c) have related code.

1. At the beginning, the program define

```
#define NO_ALERT_COMPARE (0u)
#define MILD_ALERT_COMPARE (250u)
#define HIGH_ALERT_COMPARE (500u)
```
2. `IasEventHandler` from `CyBle_IasRegisterAttrCallback(IasEventHandler);` is very important. Because in sub program `IasEventHandler`, we can see that it calls `HandleAlertLEDs(alertLevel);`
3. In `HandleAlertLEDs`, we can clearly see that it call `PWM_Write_Compare(0/250/500)` to control the PWM and then control the LED

```

161 void HandleAlertLEDs(uint8 status)
162 {
163     /* Update Alert LED status based on IAS Alert level characteristic. */
164     switch(status)
165     {
166         case NO_ALERT:
167             PWM_WriteCompare(NO_ALERT_COMPARE);
168             break;
169
170         case MILD_ALERT:
171             PWM_WriteCompare(MILD_ALERT_COMPARE);
172             break;
173
174         case HIGH_ALERT:
175             PWM_WriteCompare(HIGH_ALERT_COMPARE);
176             break;
177     }
178 }

```

We also solve a version conflict and update problem. When we use CySmart 1.2 as Bluetooth client to find PSoC 4 BLE, the CySmart 1.2 showed that the Firmware of our Dongle(dark blue board) should be upgraded. So we use PSoC Programmer 3.24.4 to upgrade Dongle's firmware by the version file named BLE_4.1_Dongle_CySmart_1.2.1.21.hex

