CYBLE-022001-00: CapSense Slider and LED

Objective

This example demonstrates how to use CYBLE-022001-00 BLE device as Peripheral device with custom profile implementation.

Overview:

This project demonstrates connectivity between the BLE Pioneer Kit (acting as a Peripheral and GATT server device) and CySmart Central Emulation tool or mobile device running the CySmart mobile application (acting as a Central and GATT client device)..

Requirements:

Programming Language	: C (GCC 4.8.4)
Associated Parts	: CYBLE-022001-00
Required software	: PSoC Creator 3.1 SP2, PSoC Programmer 3.22.3 , CySmart PC application, CySmart Android app, CySmart iOS app
Required hardware	: CY8CKIT-042-BLE Bluetooth® Low Energy (BLE) Pioneer Kit , CYBLE-022001-EVAL
Optional hardware	: Android phone with BLE support, Apple device with BLE support

Project Description:

The project source code is maintained in the GitHub location: https://github.com/cypresssemiconductorco/EZ-BLE_PRoC_Module/tree/master/Example_projects

This project demonstrates connectivity between the BLE Pioneer Kit (acting as a Peripheral and GATT server device) and CySmart Central Emulation tool or mobile device running the CySmart mobile application (acting as a Central and GATT client device). This project demonstrates following:

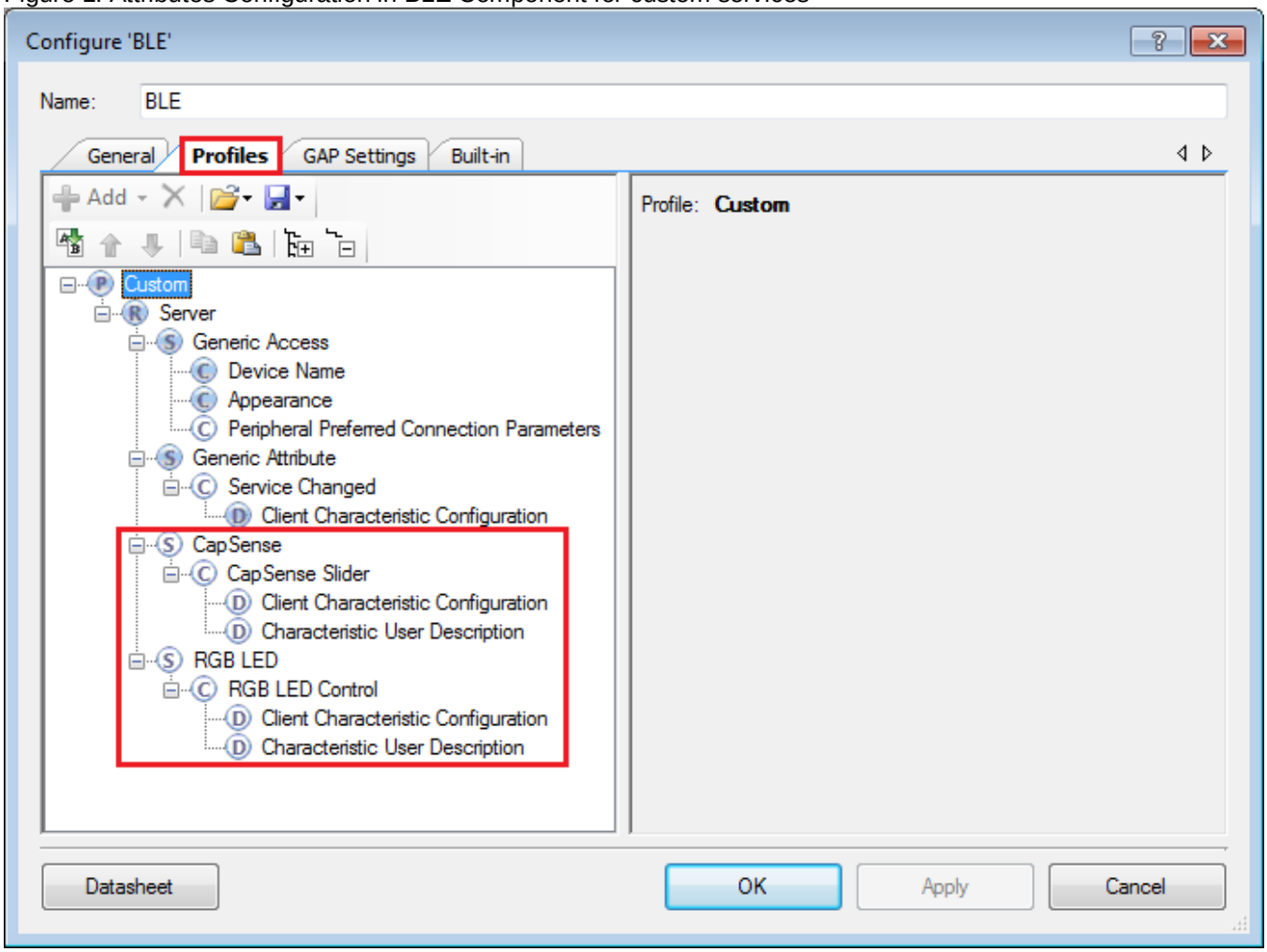
- Advertisement with timeout
- Connection with any Central device
- Two custom services in single profile
- Data transfer over BLE custom service using notifications, read, and write
- Low-power mode implementation for coin-cell operation

The BLE profile in this project consists of two BLE custom services: CapSense and RGB LED. The **CapSense service** consists of one custom characteristic, termed as CapSense Slider. The CapSense slider characteristic is used to send one byte data, ranging from 0 to 100, as notification to the GATT client device. This data is the finger location read by the CapSense component on the five-segment slider (CSS1) present on the kit. This characteristics supports notification, which allows the GATT server to send data to the connected client device whenever new data is available.

The RGB LED service also consists of one custom characteristic, termed as RGB LED Control. This characteristic supports two operations, read and write, through which the connected GATT client device can read data as well as write a new value to the characteristic. This data has four byte values indicating red, green, blue, and intensity values to control the onboard RGB LED.

These properties for the custom service/characteristics are configured in the BLE component under the Profiles tab, as shown in Figure 1 below

Figure 1. Attributes Configuration in BLE Component for custom services



The project consists of the following files:

■ **main.c/.h**

These files contain the main function, which is the entry point and execution of the firmware application. It also contains function definition for initialization of the system and reading the CapSense slider data from the CapSense component.

■ **BLEApplications.c/.h**

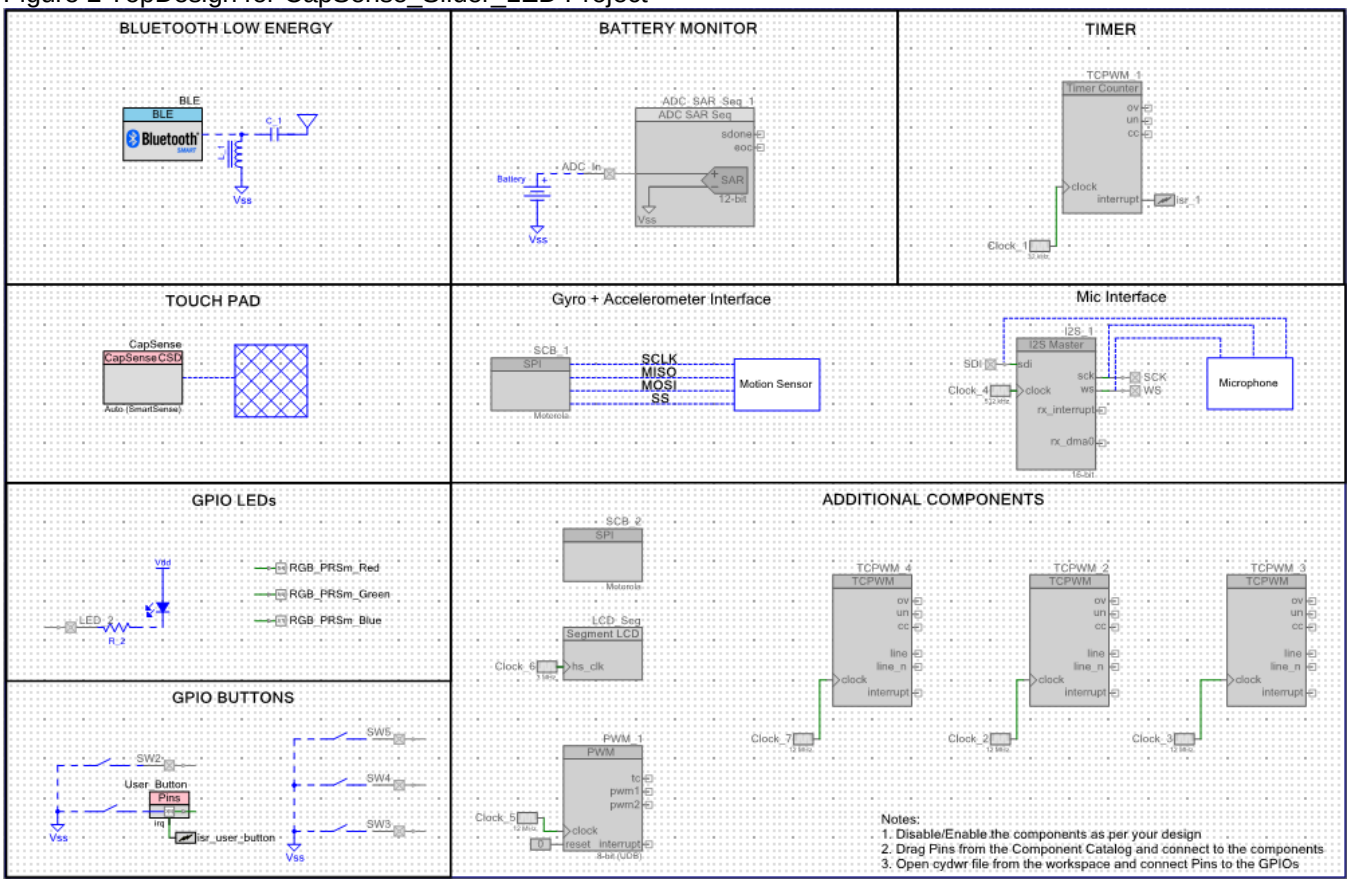
These files contain all the macros and function definitions related to BLE communication and operation. It contains the event callback function definition that is registered with the BLE component startup and used by the component to send BLE-related events from the BLE stack to the application layer for processing. It contains a method to send CapSense notifications to the GATT client device and process the Read and Write commands on the RGB LED characteristic by the GATT client device. It updates the BLE Connection parameter, which is important for low-power mode usage.

■ **HandleLowPower.c/.h**

These files contain the function to handle low-power mode. This function is continuously called in the main loop and is responsible for pushing the BLE hardware block (BLESS) as well as the CPU to Deep Sleep mode as

much as possible. The wakeup source is either the BLE hardware block Link Layer internal timer or the interrupt from the user button press (**SW2**). This allows for very low power mode implementation and operation using a coin cell.

Figure 2 TopDesign for CapSense_Slider_LED Project



Hardware Connections

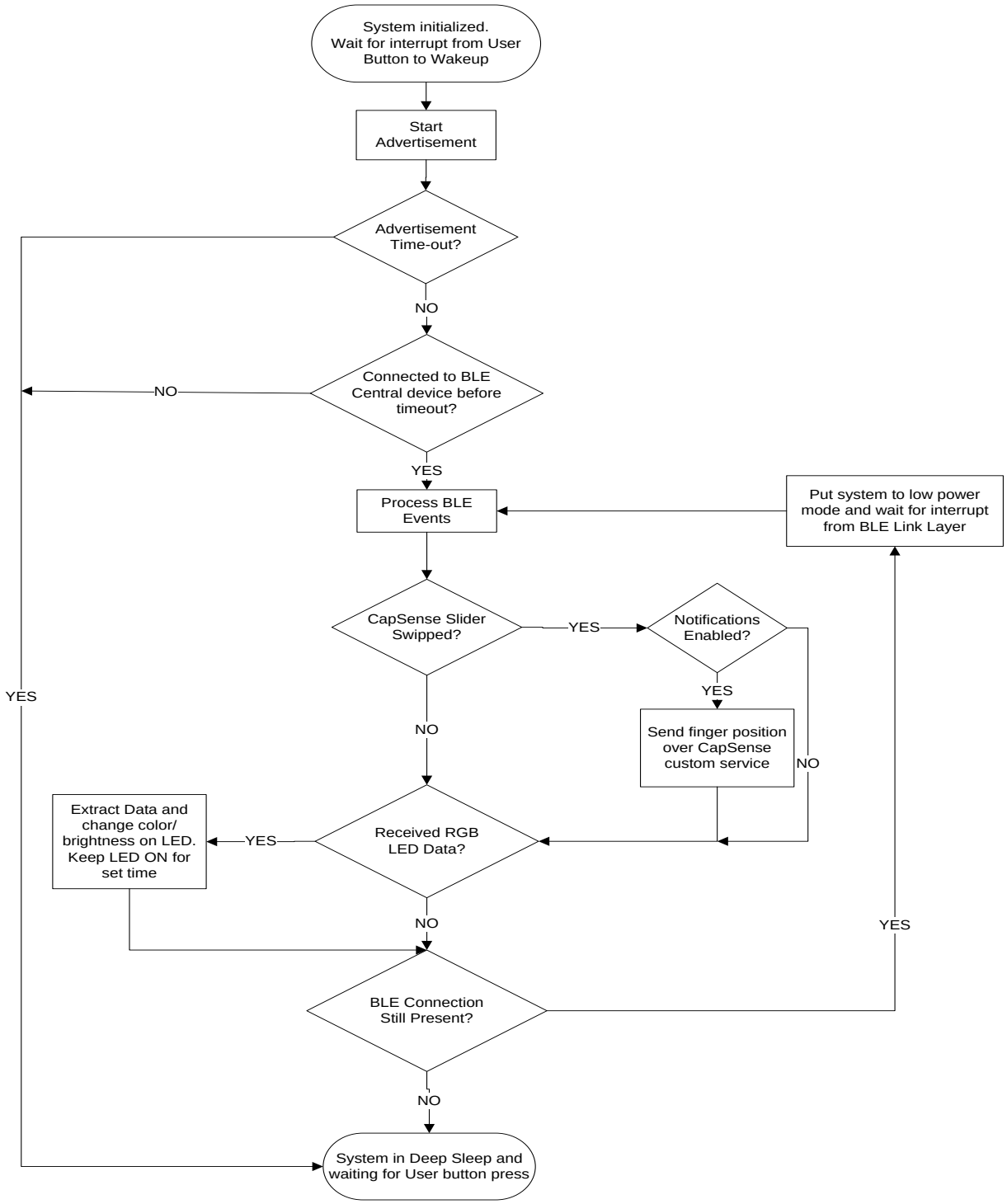
No specific hardware connections are required for this project because all connections are hardwired on the BLE Pioneer Baseboard. Ensure that the module is placed on the baseboard correctly.
The pin assignment for this project is in **CapSense_Slider_LED.cydwr** in the Workspace Explorer, as shown in Figure 3

Figure 3. Pin Selection for CapSense Slider and LED Project

Alias	Name	Port	Pin	Lock
Cmod	\CapSense:Cmod\	P4[0] CSD:c_mod, TCPWM0:line_out, SCB1:uart rts, SCB1:spi mosi	24	☑
LinearSlider0_e0__LS	\CapSense:Sns[0]\	P4[1] CSD:c_sh_tank, TCPWM0:line_out_compl, SCB1:uart cts, SCB1:spi miso	31	☑
LinearSlider0_e1__LS	\CapSense:Sns[1]\	P1[7] OA3:vplus_alt, TCPWM3:line_out_compl, SCB0:uart cts, SCB0:spi clk	25	☑
LinearSlider0_e2__LS	\CapSense:Sns[2]\	P1[6] OA2:vplus_alt, TCPWM3:line_out, SCB0:uart rts, SCB0:spi select[0]	37	☑
LinearSlider0_e3__LS	\CapSense:Sns[3]\	P0[5] LPCOMP:in_n[1], TCPWM1:line_out_compl, SCB0:uart_tx, SCB0:i2c_scl, SCB0:spi miso	57	☑
LinearSlider0_e4__LS	\CapSense:Sns[4]\	P0[4] LPCOMP:in_p[1], TCPWM1:line_out, SCB0:uart_rx, SRSS:ext_clk, SCB0:i2c_sda, SCB0:spi mosi	64	☑
	RGB_PRSm_Blue	P3[7] SARMUX:pads[7], TCPWM3:line_out_compl, SCB1:uart cts, SRSS:ext clk lf	4	☑
	RGB_PRSm_Green	P3[6] SARMUX:pads[6], TCPWM3:line_out, SCB1:uart rts	14	☑
	RGB_PRSm_Red	P3[4] SARMUX:pads[4], TCPWM2:line_out, SCB1:uart rx, SCB1:i2c sda	12	☑
	User_Button	P3[5] SARMUX:pads[5], TCPWM2:line_out_compl, SCB1:uart tx, SCB1:i2c scl	13	☑

Flow chart

Figure 4 shows the flow chart of the code implemented.


System initialized. Wait for interrupt from User Button to Wakeup
Start Advertisement
Advertisement Time-out?
NO
Connected to BLE Central device before timeout?
NO
YES
Process BLE Events
Put system to low power mode and wait for interrupt from BLE Link Layer
CapSense Slider Swipped?
YES
Notifications Enabled?
YES
Send finger position over CapSense custom service
NO
NO
YES
Extract Data and change color/ brightness on LED. Keep LED ON for set time
YES
Received RGB LED Data?
NO
YES
BLE Connection Still Present?
NO
System in Deep Sleep and waiting for User button press

Verify Output

The project can be verified by two methods: using the CySmart Central Emulation Tool and BLE Dongle or using the CySmart mobile application.

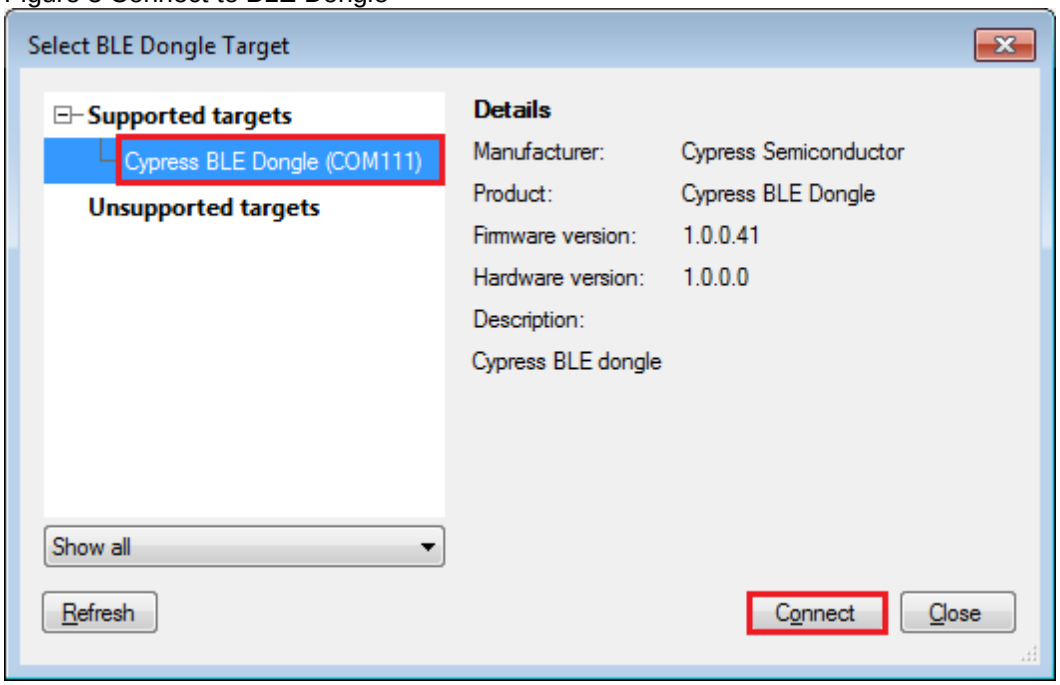
CySmart Central Emulation Tool

To verify the CapSense and LED project using the CySmart Central Emulation Tool, follow these steps:
Note: Refer CySmart BLE Host Emulation tool to learn how to use the tool.

1. Connect the BLE Dongle to one of the USB ports on the PC.
2. Start the CySmart Central Emulation Tool on the PC by going to **Start > All Programs > Cypress> CySmart <version> > CySmart <version>**. You will see a list of BLE Dongles connected to it.

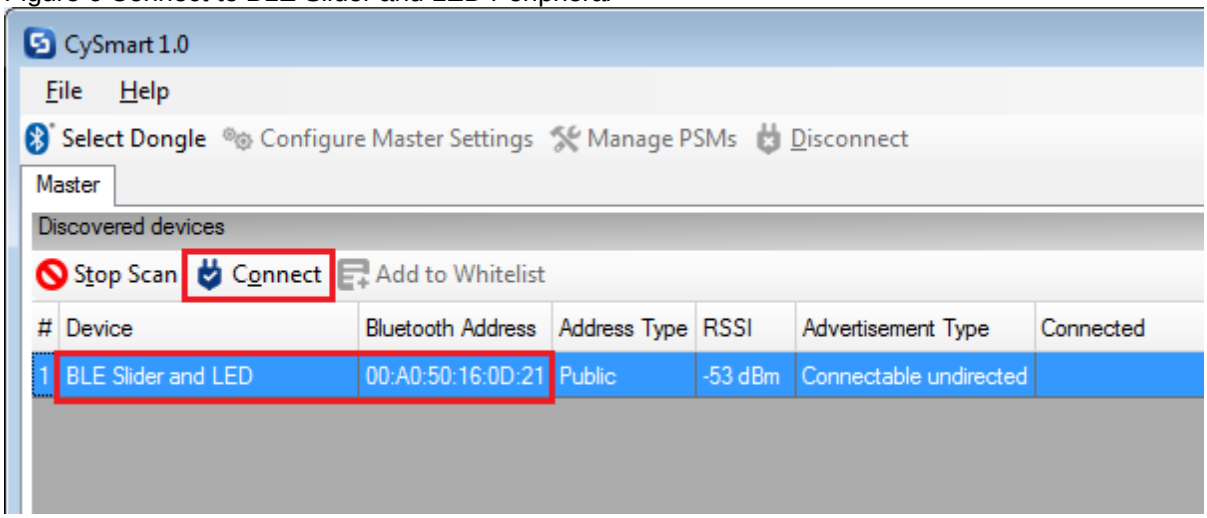
If no BLE Dongle is found, click **Refresh**. Select the BLE Dongle and click **Connect**.

Figure 5 Connect to BLE Dongle



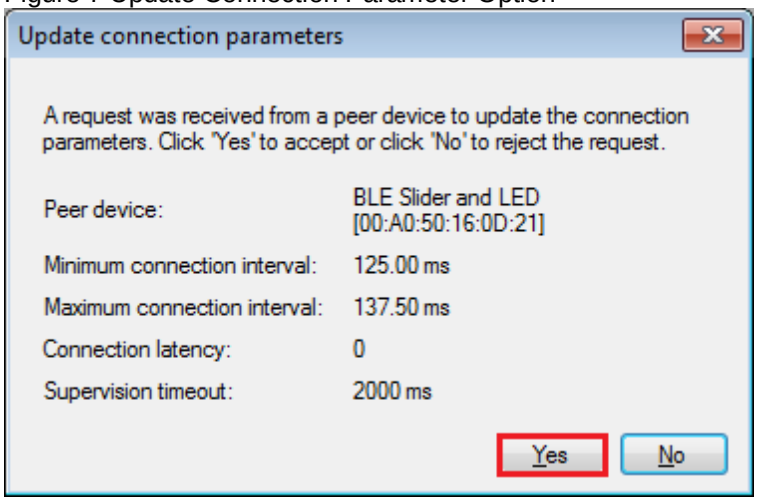
3. Place module on the BLE Pioneer Kit, depending on the project chosen.
4. Power the BLE Pioneer Kit through the USB connector **J13**.
5. Open the project in creator and buid it.Program the BLE Pioneer Kit with the CapSense and LED example project.
6. After programming successfully, press the user button (**SW2**) on the BLE Pioneer kit to start the advertisement. Advertisement is indicated by a blinking red LED on the baseboard.
Note: The project has an advertisement timeout of 30 seconds after which it returns to Deep Sleep. Press **SW2** again to restart the advertisement.
7. On the CySmart Central Emulation Tool, click **Start Scan** to see the list of available BLE peripheral devices.
8. Double-click the **BLE Slider and LED** device to connect, or click **BLE Slider and LED** and then Click **Connect**.

Figure 6 Connect to BLE Slider and LED Peripheral



When connected, the CySmart Central Emulation Tool will display a message for the **Update connection parameters**. Select **Yes**, as shown in Figure 7

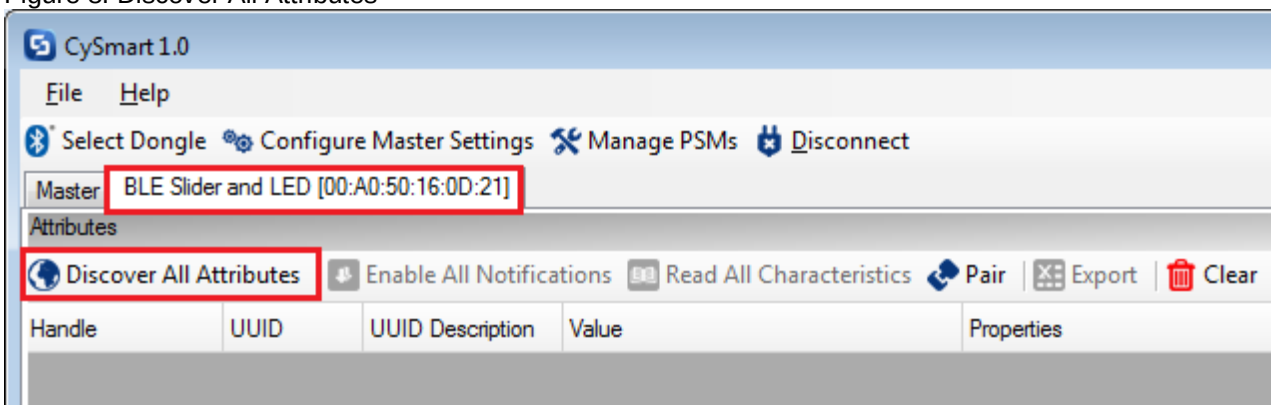
Figure 7 Update Connection Parameter Option



Note: If you select **No**, the project will still work. However, the current consumption will be higher due to faster connection interval.

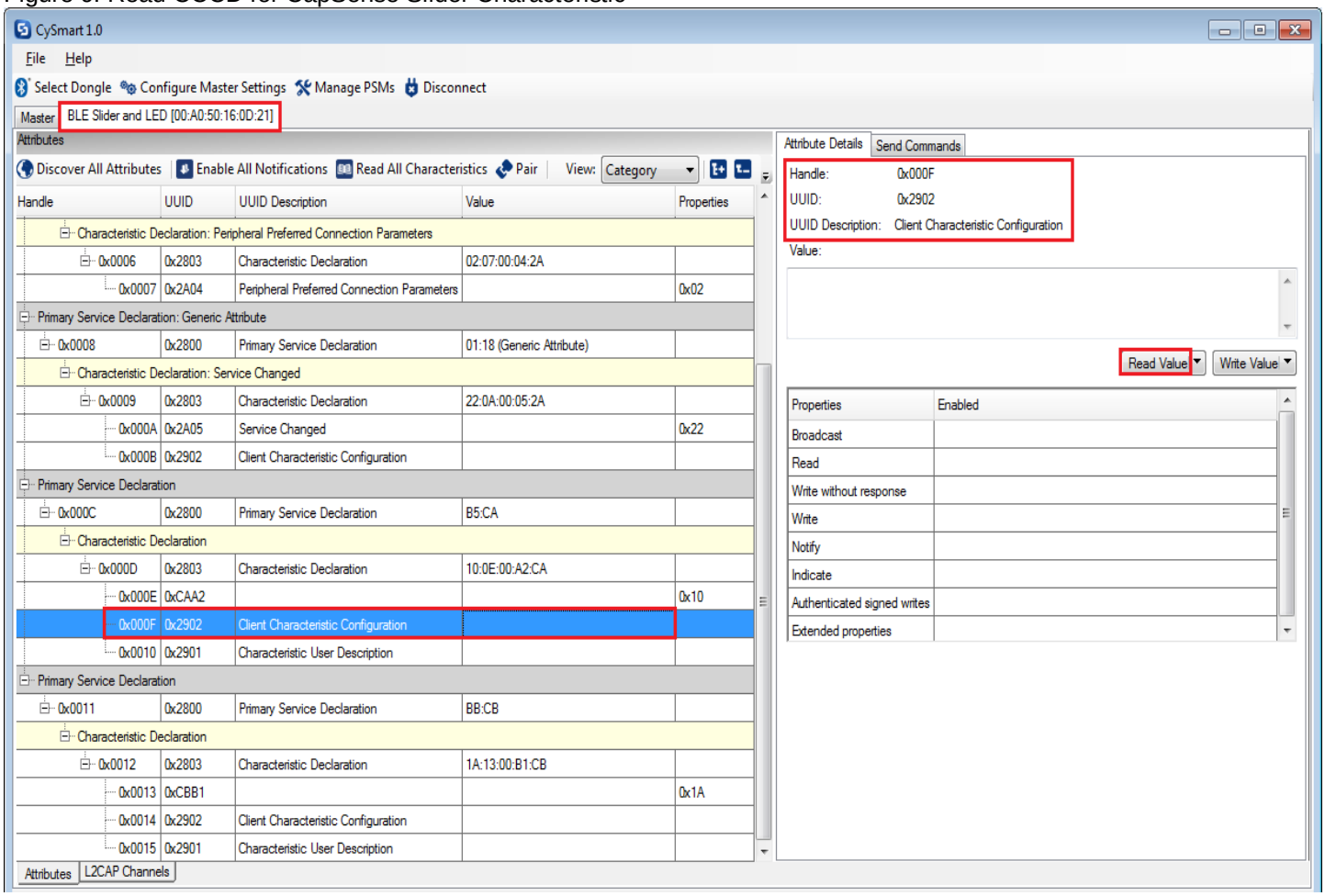
10. Click **Discover All Attributes** to find all attributes supported.

Figure 8. Discover All Attributes



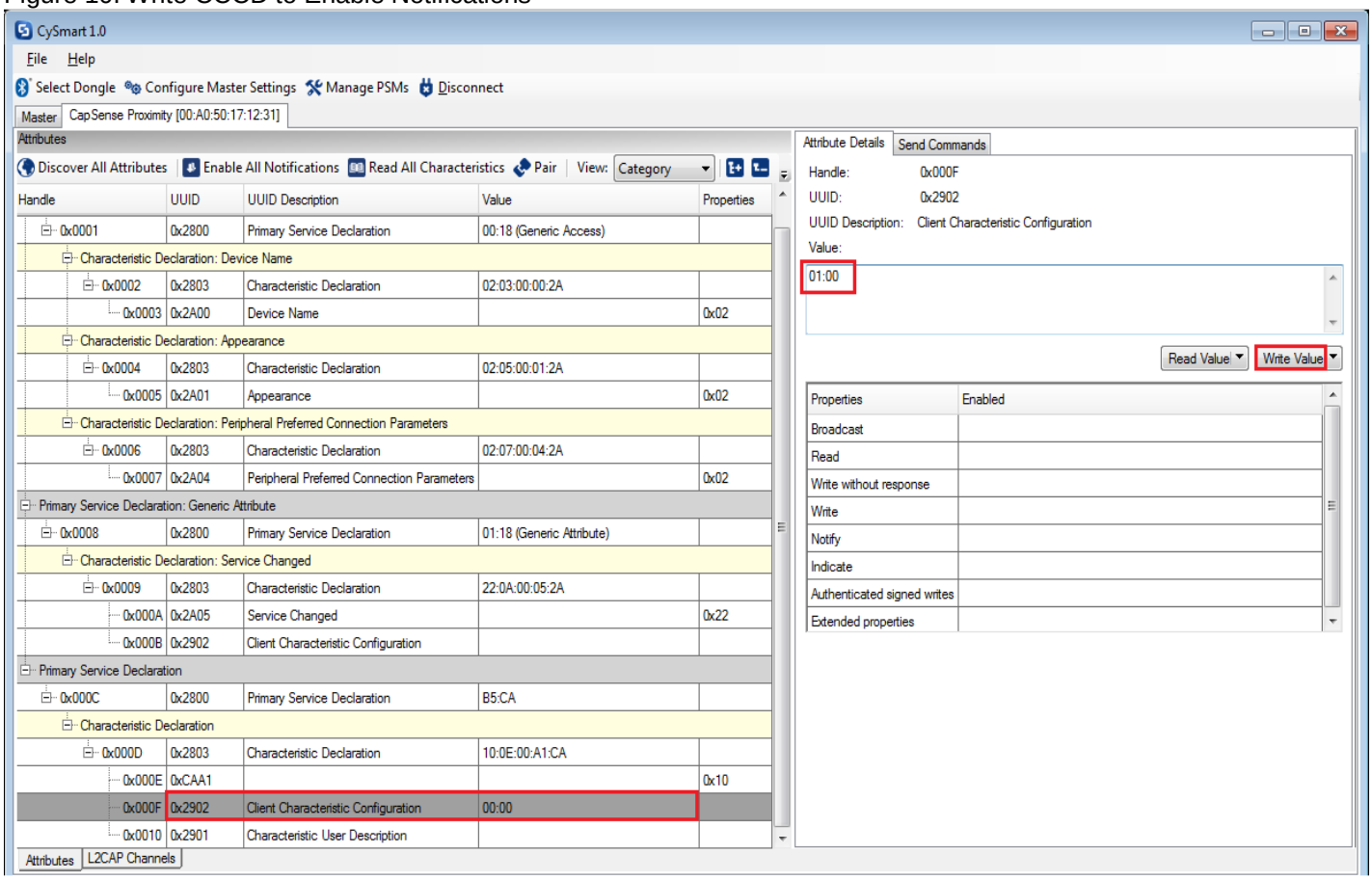
11. Locate the attribute **Client Characteristic Configuration** descriptor (UUID 0x2902) under CapSense Slider characteristic (**UUID 0xCAA2**). Click **Read Value** to read the existing Client Characteristic Configuration Descriptor (CCCD) value as shown in Figure 9.

Figure 9. Read CCCD for CapSense Slider Characteristic



12. Modify the **Value** field of CCCD to '01:00' and click **Write Value**. This enables the notifications on the **CapSense Slider** characteristic.

Figure 10. Write CCCD to Enable Notifications



13. Swipe your finger on the CapSense slider on the BLE Pioneer kit, as shown in Figure 11 and see the notification values in the CapSense Slider value field, as shown in Figure 12.

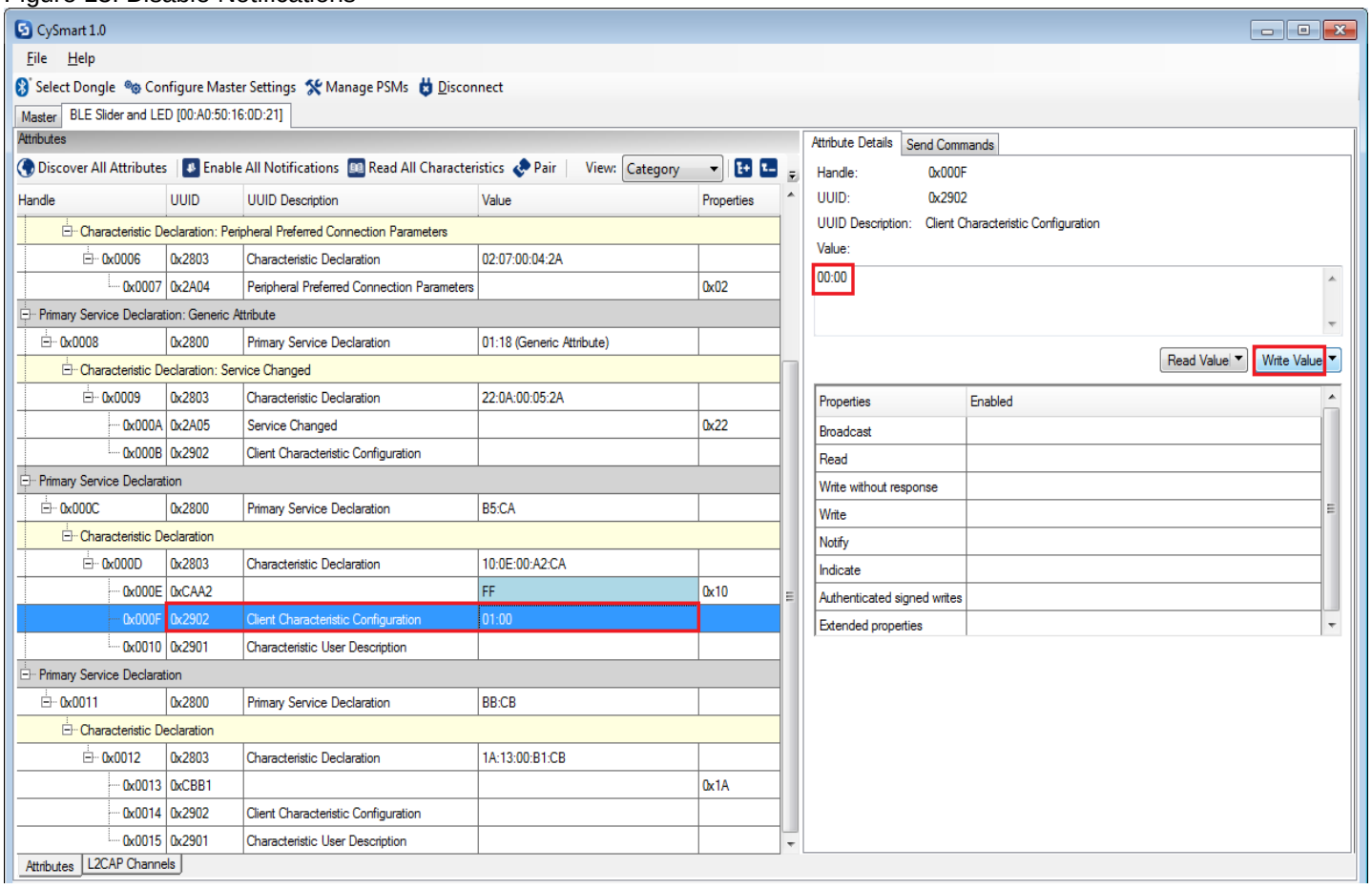
Figure 11. CapSense Slider

Figure 12. CapSense Slider Notification Received

Primary Service Declaration				
0x000C	0x2800	Primary Service Declaration	B5:CA	
Characteristic Declaration				
0x000D	0x2803	Characteristic Declaration	10:0E:00:A2:CA	
0x000E	0xCAA2		44	0x10
0x000F	0x2902	Client Characteristic Configuration	01:00	
0x0010	0x2901	Characteristic User Description		

14. To disable notifications, modify the **Value** field of the **Client Characteristic Configuration** descriptor to '00:00' and click **Write Value**.

Figure 13. Disable Notifications



15. Locate the **RGB LED Control** characteristic (**UUID 0xCBB1**). Click **Read Value** to read the existing 4-byte onboard RGB LED color information, as shown in Figure 14.

Figure 14. Read RGB LED Control Characteristic Value

Primary Service Declaration				
0x0011	0x2800	Primary Service Declaration	BB:CB	
Characteristic Declaration				
0x0012	0x2803	Characteristic Declaration	1A:13:00:B1:CB	
0x0013	0xCBB1		00:00:00:00	0x1A
0x0014	0x2902	Client Characteristic Configuration		
0x0015	0x2901	Characteristic User Description		

16. Modify the four bytes of data in the **Value** field and click **Write Value**. You will see the corresponding change in the color and intensity of the RGB LED on the BLE Pioneer kit, as shown in Figure 15. The RGB LED will be on for 3 seconds before switching off to conserve power.

Note: If the BLE Pioneer kit is powered from a coin cell and not the USB Vbus, then the color mixing and intensity will vary. This is because the coin cell provides a lower driving voltage for RGB LEDs.

Figure 15. Write RGB LED Control Characteristic Value

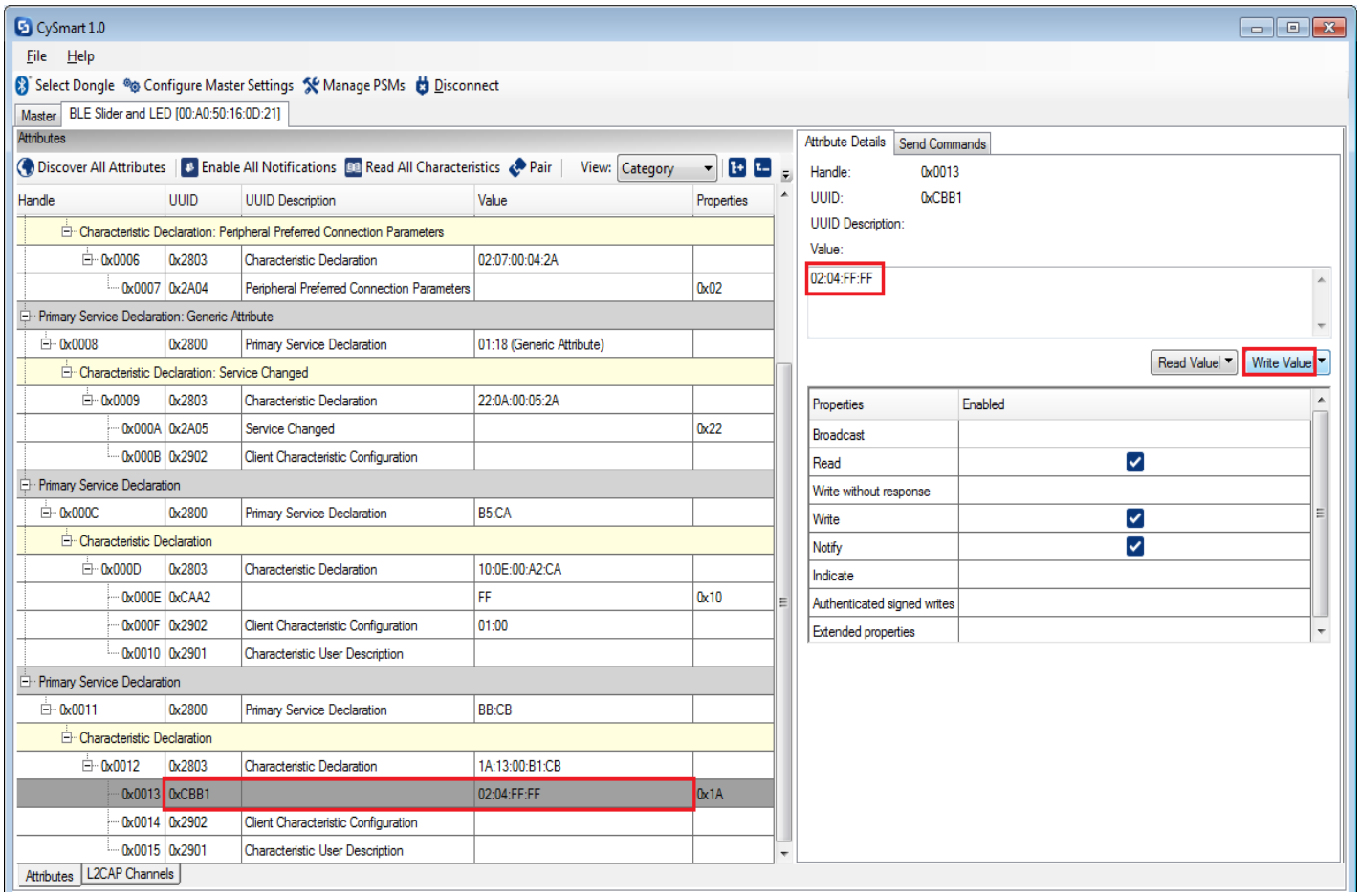
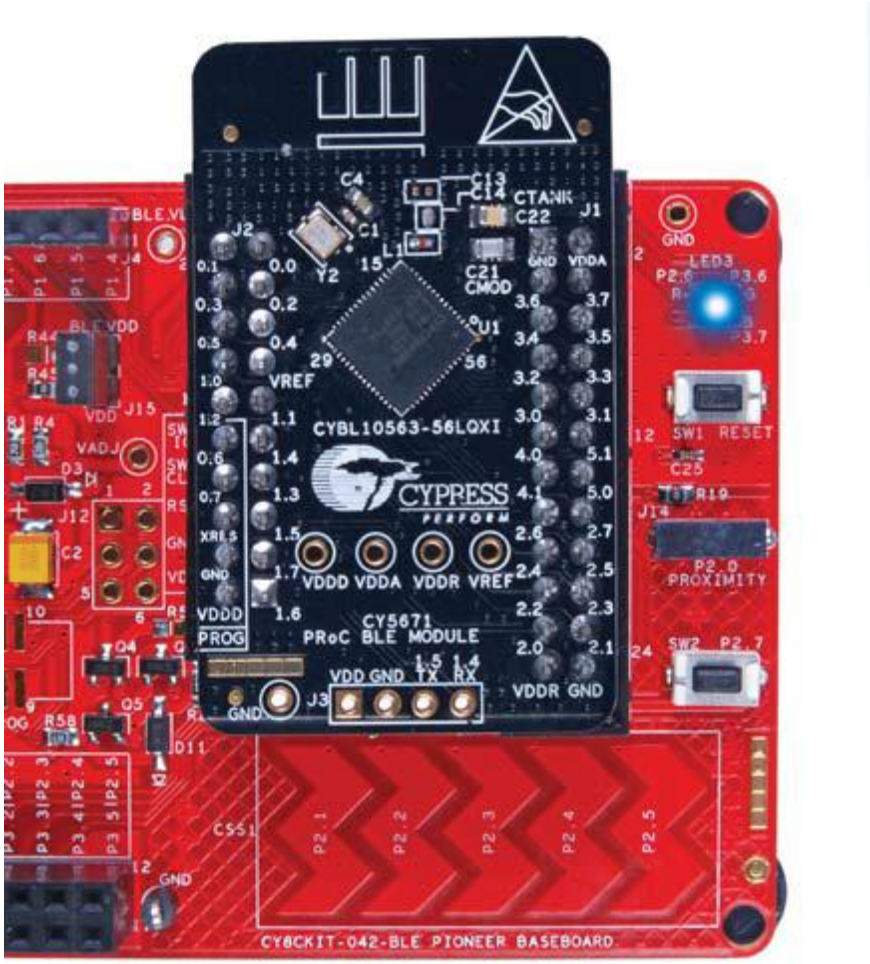
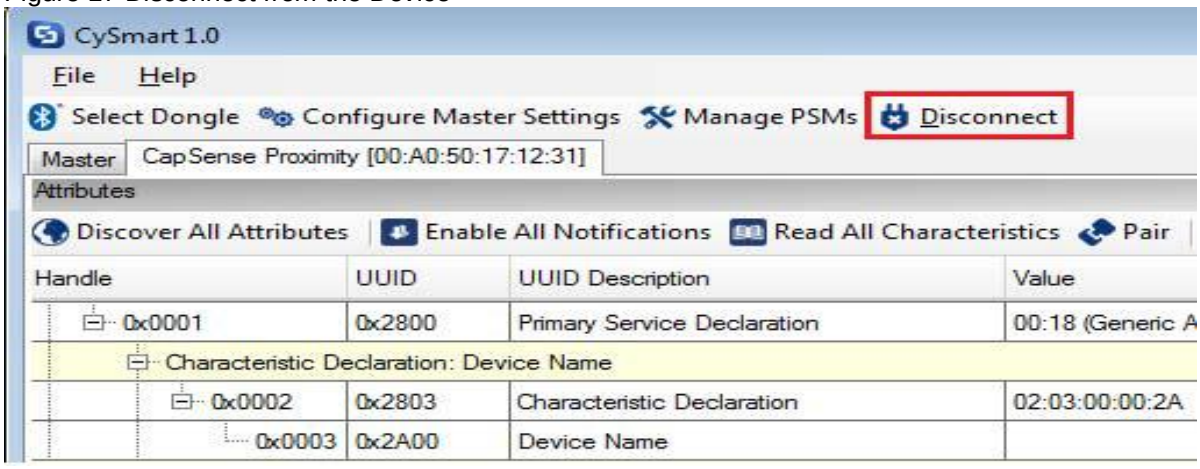


Figure 16. RGB LED Control with PSoC 4 BLE Module and PRoC BLE Module



17. To disconnect from the device, click **Disconnect**, as shown in Figure 17

Figure 17 Disconnect from the Device

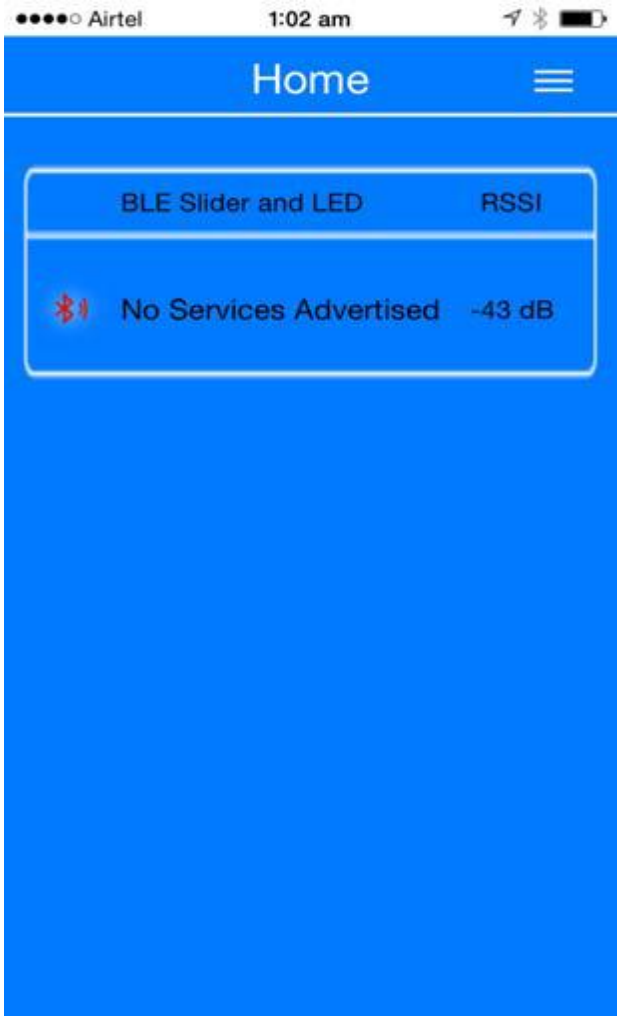


18. To connect to this Peripheral again, restart advertising by pressing the user button (**SW2**) on the BLE Pioneer Kit. Advertising is indicated by the blinking red LED.

CySmart Mobile Application

To verify the CapSense and LED project using the CySmart mobile application, follow these steps:
1. Connect the CYBLE-022001-EVAL board on the J10 and J11 headers on the BLE Pioneer Kit.
2. Connect the BLE Pioneer Kit into the PC using the J13 USB connector.
3. Program the kit with the CapSense and LED example projects.
4. Press the user button (**SW2**) on the BLE Pioneer Kit to start the advertisement. This is indicated by the blinking red LED on the BLE Pioneer Kit.
5. Open the application on the mobile device. If Bluetooth is not enabled on the device, the application will ask to enable it.
6. After Bluetooth is enabled, the CySmart mobile application will automatically search for available Peripherals and list them. Select the **BLE Slider and LED** Peripheral as shown in Figure 18.

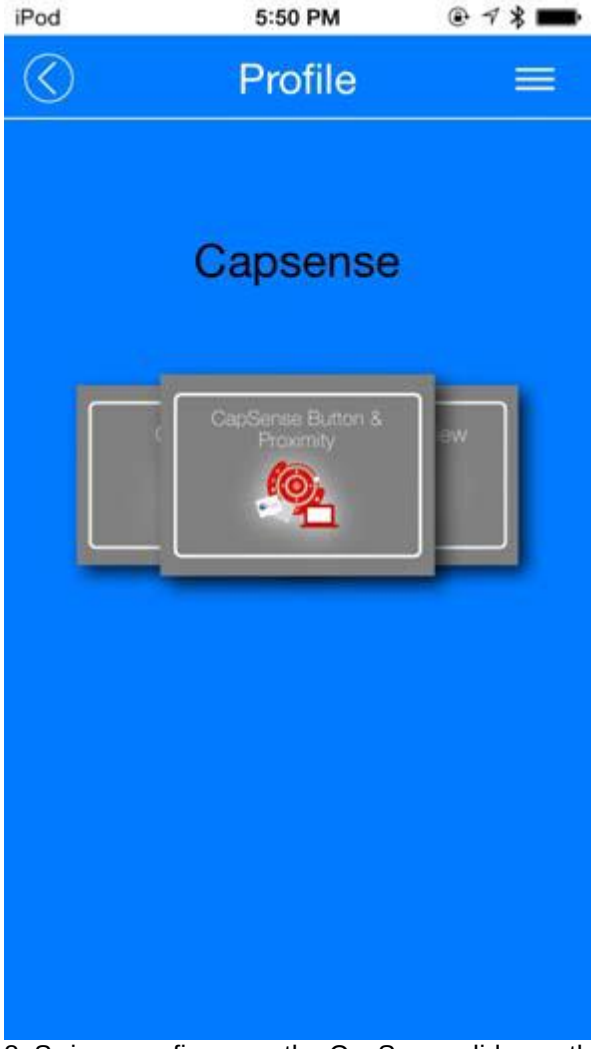
Figure 18. BLE Slider and LED Peripheral



7. When connected, the CySmart mobile application will list the supported profiles by the Peripherals.

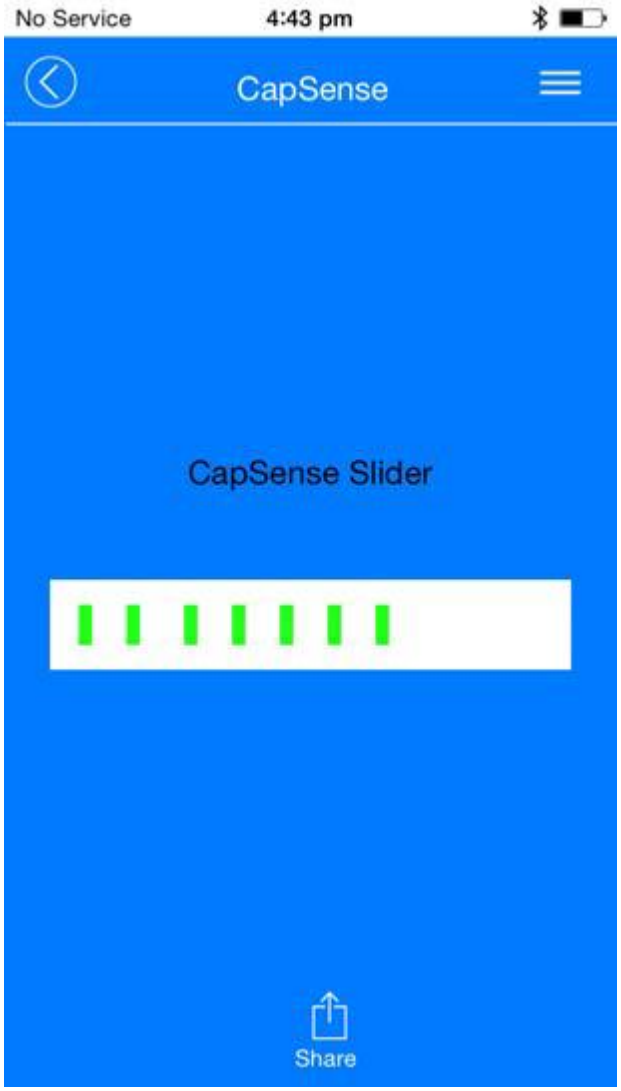
Scroll and select the CapSense icon, as shown in Figure 19.

Figure 19. CapSense Service Page



8. Swipe your finger on the CapSense slider on the BLE Pioneer Kit and see a similar response on The **CapSense** page in the CySmart application Figure 20.

Figure 20. CapSense Slider



9. Press the back button to return to the service selection page. Scroll and tap on the RGB LED service.
10. On the RGB LED service page, swipe over the color gamut to see a similar color response on the BLE Pioneer Kit RGB LED. The slider below the color gamut controls the intensity of the RGB LED color. The RGB LED will be on for 3 seconds before switching off. This is done to conserve power.

Figure 21 RGB LED Control with CySmart mobile application

11. To disconnect from the BLE Pioneer Kit, return to the CySmart mobile application home screen by pressing the back button.
12. To reconnect to the peripheral, press the user button (**SW2**) on the BLE Pioneer Kit again and then scan for devices using CySmart mobile application.