PSoC4 BLE -Central and Peripheral

Objective

This example demonstrates how to use PSoC4 BLE as both BLE **Central and Peripheral.**

Overview

In this example BLE component is used to configure PSoC4 BLE in both Central and Peripheral GAP role. In this example project, when device is acting as Central, it can scan and connect to peer device which is advertising. When acting as peripheral it can advertise and other Central can connect to the PSoC4 BLE. The project is controlled based on the commands from UART, accordingly changing PSoC4 BLE role

Requirements

Tool: PSoC Creator 3.1, CySmart 1.0

Programming Language: C (GCC 4.8.4)

Associated Parts: All PSoC 4 BLE parts

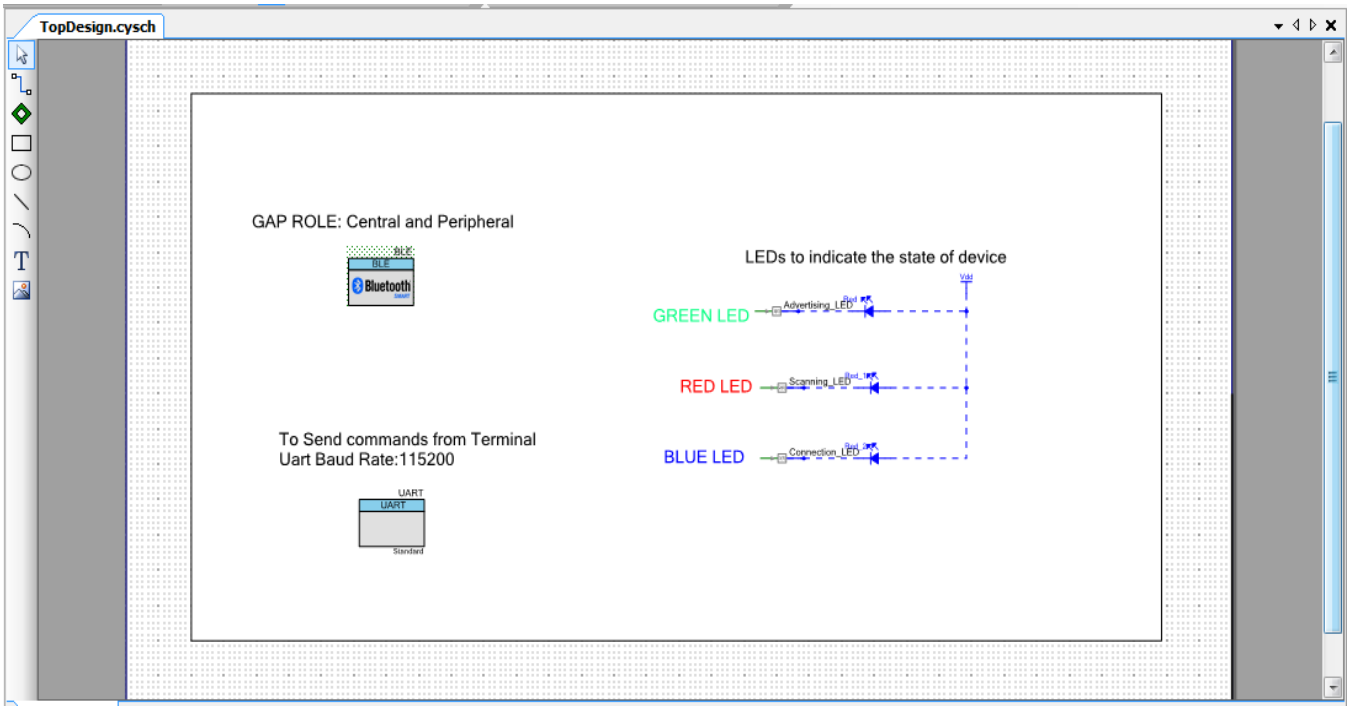
Related Hardware: CY8CKIT-042-BLE Bluetooth® Low Energy (BLE) Pioneer Kit

Hardware Setup

BLE Pioneer Kit has all the necessary hardware connections required for this lab. If you are using your own hardware, then connect the LEDs to P3.7, P3.6 and P2.6 of PSoC4 BLE.

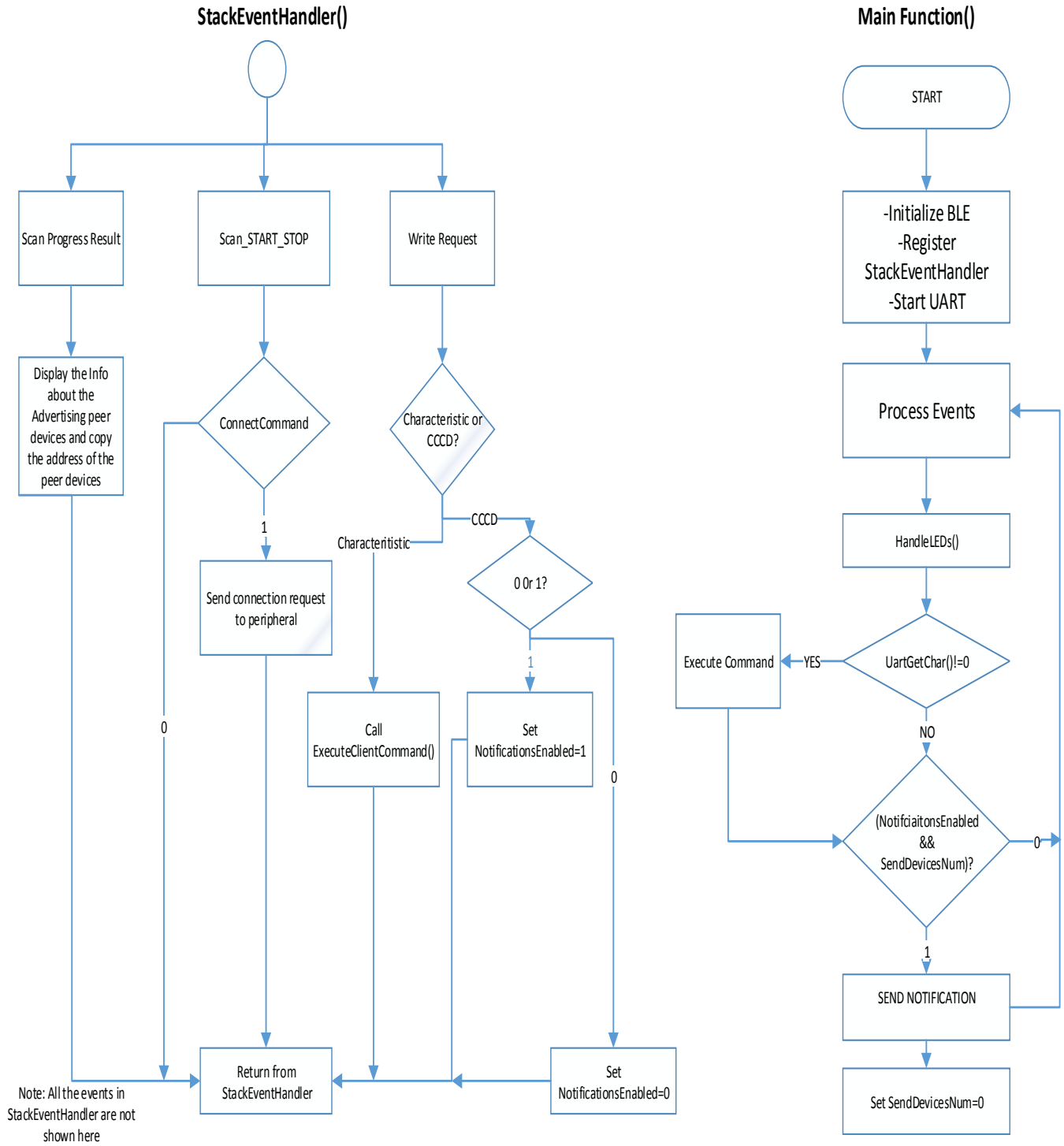
PSoC Creator Schematic

Figure 1. PSoC Creator Schematic



Operation:

Figure 2(a). Firmware Flow



1. **main() function**: This is the function which performs the
   - Initialization of the BLE stack
   - Executes the necessary routines to process the BLE events
   - Receive and execute commands from UART terminal and send notificaitions/indications.
2. In the initial section of the *main()* function, the API *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 assume any name – in this project, we used *StackEventHandler*. Once the system is initialized, *main()* function continuously operates in a *for(;;)* loop executing *CyBle_ProcessEvents(),*UART terminal commands and will send notifications (if enabled). CyBle_ProcessEvents() processes the events received by the BLE stack and enables application layer to use them and take the appropriate action. Commands from UART terminal

   i. 'a' ->Start advertising if device is in **DISCONNECTED** state .Stop scanning and start advertising if device is in the state **SCANNING**.
   ii. 's'->Start scanning if device is in **DISCONNECTED** state. Stop advertising and start scanning if the device is in the state **ADVERTISING**.
   iii. 'z'->to select the peer devices among the received advertising reports during scanning
   iv. 'c'->Stop scanning and connect to the selected peer device
   v. 'd'->Disconnect the connection
   vi. 'u'->Send Update connection parameters request/command(peripheral/central).

3. **StackEventHandler() function**: This function handles the common events generated for the BLE Stack.

I. `CYBLE_EVT_GAPC_SCAN_PROGRESS_RESULT:` This event will occur when it receives any advertising packet or scan response data. In this event we are copying the address of the peer advertising devices.

II. `CYBLE_EVT_GAPC_SCAN_START_STOP:` This event occurs when scanning is started or stopped. If the scanning is stopped to connect to some device then send connection request to that particular device or if the scanning is stopped to start advertising then start advertising and set Stop=0.

III. `CYBLE_EVT_L2CAP_CONN_PARAM_UPDATE_RSP:` This event will occur when slave receives L2CAP connection update response. Based on the eventParam we will get to know whether the request is accepted or rejected.

IV. `CYBLE_EVT_GATTS_WRITE_REQ:` These events will trigger when the client sends a s write request or write command to the server. Client will enable or disable the notifications by writing to CCCD. Similarly Client can write some value to the characteristic **.**

   I. If client writes 0x01 device starts to scan and after scan timeout it will send the number of peer devices which are advertising during that time to the client.

4. **ExecuteClientCommand() function**: This function is used to start scan when client writes 0x01 to the characteristic and set the variable **`SendDevicesNum`**`=1;`.
5. **HandleLeds()** function: This function is called in the while(1) in main().The function TURNs ON or TURN OFF the LEDS based on the current state of the device.

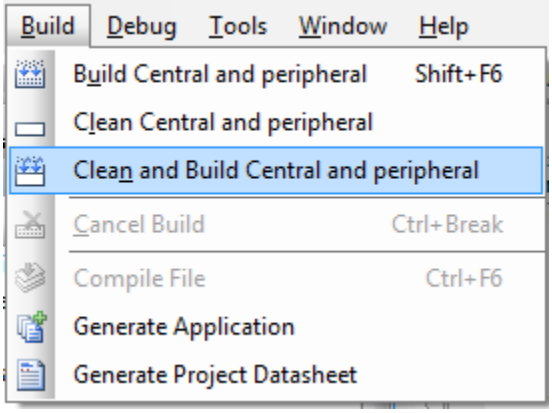
Build and Program the device

This section shows how to build the project and program the PSoC4 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.

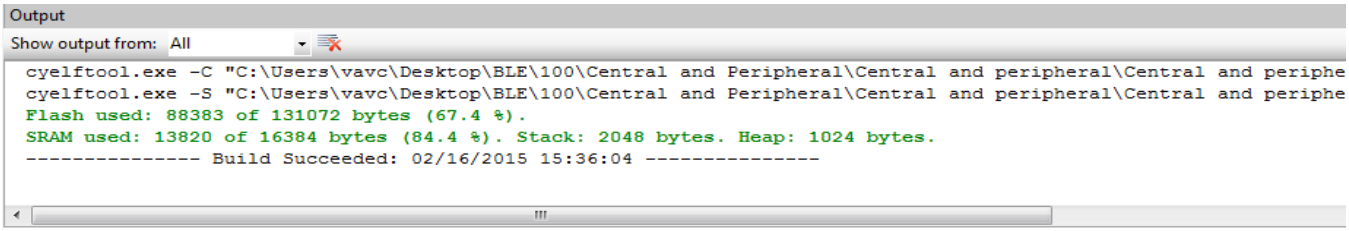
1. On PSoC Creator, select **Build** > **Clean and Build Central and Peripheral**, as shown in Figure 3.

Figure 3. Build Project



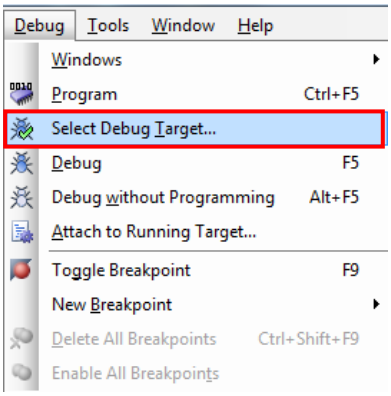
2. On successful build, total flash and SRAM usage is reported, as shown in Figure 4.

Figure 4. Build Succeeded



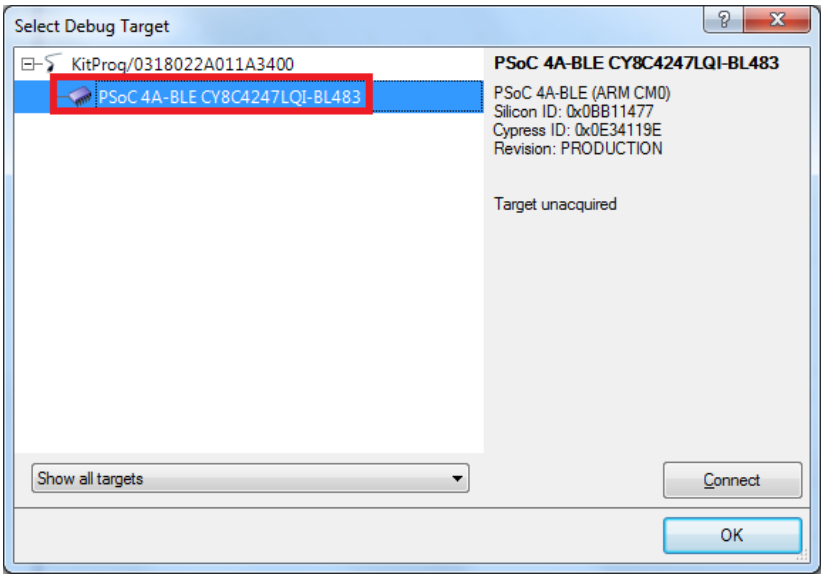
3. Select **Debug** > **Select Debug Target**, as shown in Figure 5.

Figure 5. Selecting Debug Target



4. 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.

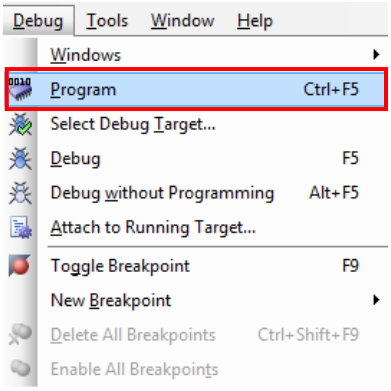
Figure 6. Connecting to a Device



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.

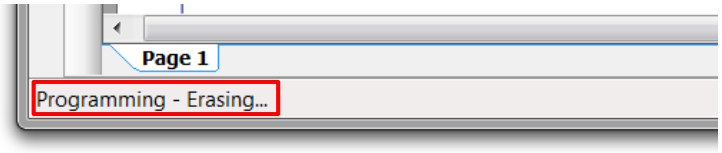
5. Select **Debug** > **Program** to program the device with the project, as shown in Figure 7.

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



Steps to test the project:

a. Open UART Terminal and the select the COM port (related to PSOC4 BLE device) and use the following settings

I. Baud rate:115200

II. Data rate: 8 bit
III. Parity: none
IV. Stop: 1 bit
V. Flow control : none

b. After the device is programmed a message "**BLE STACK ON**" will be displayed on UART terminal.

c. Connect to the peer device as master:

- After the device is programmed send command "**s**" to scan for the peer devices. While scanning RED LED will be turned **ON**
- The advertising data and scan response data will be displayed in the UART terminal during scanning
- Send command "**z**" to select the particular device in the list of devices that is displayed on UART terminal
- Send command '**c'** to connect to the peer device selected after sending the command 'z'. If no device is selecting by using '**z'** then by default it will connect to the first device

d. Connect to the peer device as slave:

- After device is programmed send command "**a**" to start to advertise. Use Dongle or any central device to connect the device (central and peripheral).

e. To connect to a new device (either as master or slave) the current connection has to be disconnected and establish a new connection .To disconnect send the command "**d**" from the UART terminal.

f. After disconnection you can either start to scan or advertise.

g. **Scanning while maintaining connection**:

- After the device is connected client has to write **0x01** to the CCCD descriptor to enable the notifications.
- Write **0x01** to custom characteristic to start scanning. After SCAN time OUT it will notify the client about the number of advertising reports received from different peer devices.

Expected Results:

The following things will be displayed on UART terminal during the execution.

BLE STACK ON:

Device entered General Discovery mode.

Started to Advertise

Device connected
Connected as Slave

Notifications enabled
Received command from Client:START SCANNING

Started To scan
CYBLE_EVT_GAPC_SCAN_PROGRESS_RESULT:
eventType:Connectable undirected advertising

peerAddrType: PUBLIC

peerBdAddr: 00a050000009

CYBLE_EVT_GAPC_SCAN_PROGRESS_RESULT:
eventType:SCAN_RSP

peerAddrType: PUBLIC
peerBdAddr: 00a050000009

Stopped scanning
DeviceNearBy: 1
Sent notification to Client

Update connection parameters request sent successfully

new connection parameters are accepted by master
Disconnect Device API Success
Device disconnected
CyBle_GapcStartScan API Success
Started to Scan
CYBLE_EVT_GAPC_SCAN_PROGRESS_RESULT:
eventType:Connectable undirected advertising

peerAddrType: PUBLIC
peerBdAddr: 00a0500d041a

CYBLE_EVT_GAPC_SCAN_PROGRESS_RESULT:
eventType:SCAN_RSP

peerAddrType: PUBLIC
peerBdAddr: 00a0500d041a

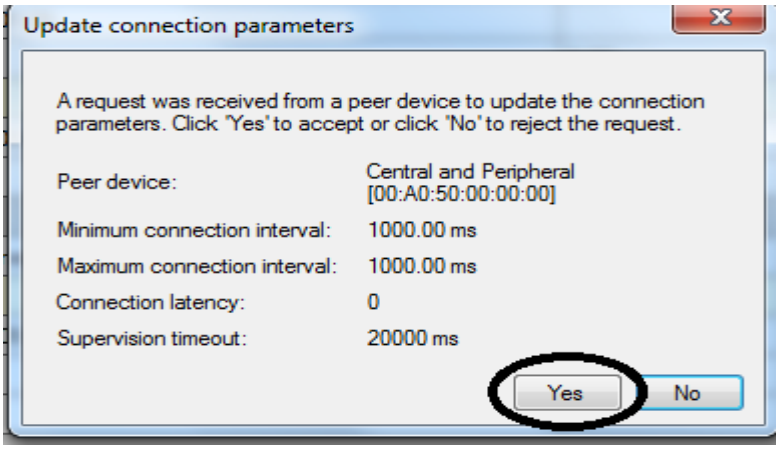
Device 0: 00a0500d041a
Select Device:0
send 'c' to connect to the selected device

Stop Scanning ...
Stopped Scanning
Connected to device with address - 00a0500d041a
Device connected
Connected as Master

To get the above results follow the sequence:

1. After programming send ‘a’ to start advertising .Use dongle and connect to the device. Write 0x01 to the CCCD of custom characteristic to enable notifications and write 0x01 to custom characteristic to start scanning.
2. Send ‘u’ to update the connection request. In the Cysmart, select YES in the pop-up window that appears to update the connection parameters.

Figure 9.Connection parameters update



3. Send ‘d’ to disconnect the connection
4. Send ‘s’ to start scanning and then send ‘z’ to select the peer device and then send ‘c’ to connect to the device.

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 PSoC4 BLE	Provides an introduction to PSoC4 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.