PSoC4 BLE -Observer

Objective

This example demonstrates how to use PSoC4 BLE device as BLE observer.

Overview

This code example uses the BLE component to demonstrate the Observer role of PSoC 4 BLE device. In this code example, device scans for the nearby advertising devices continuously and shows the advertising report and scan response data of the advertising devices on the UART terminal (like Tera Term, Hyperterminal).

Requirements

Tool: PSoC Creator 3.1,

Programming Language: C (GCC 4.8.4)

Associated Parts: All PSoC4 BLE parts

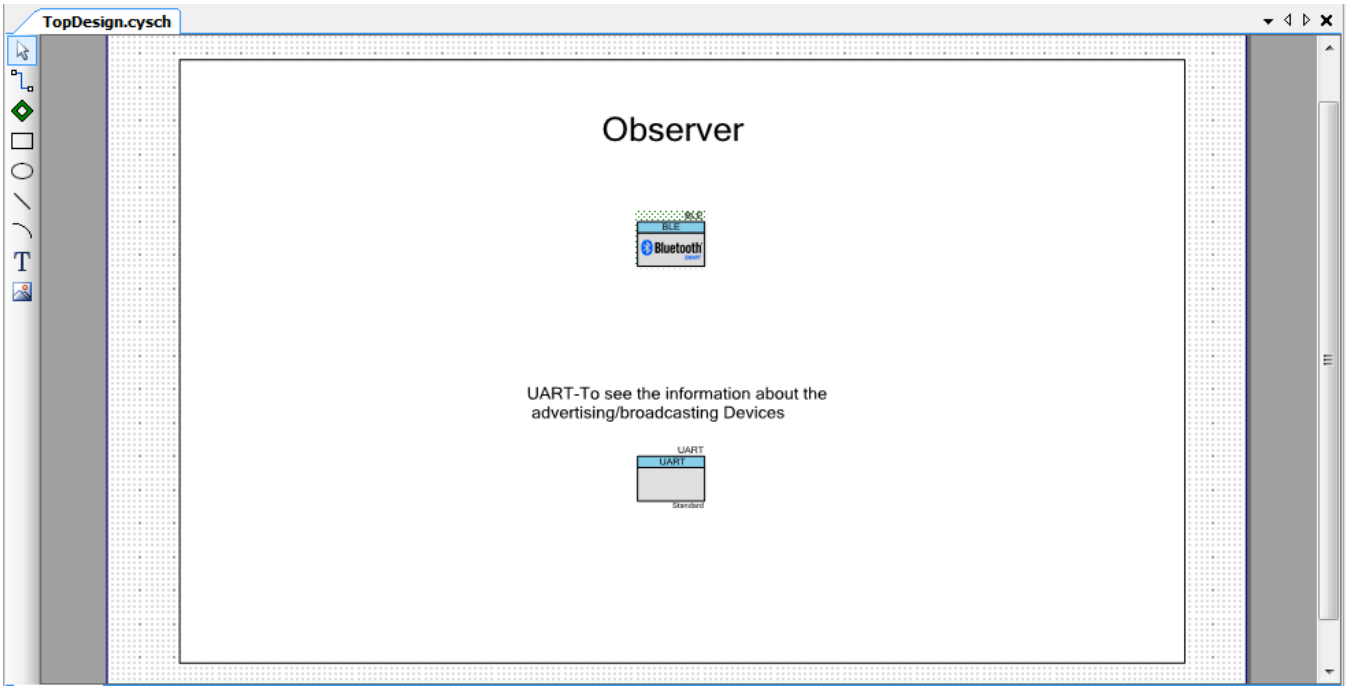
Related Hardware: CY8CKIT-042-BLE Bluetooth® Low Energy (BLE) Pioneer Kit

Hardware Setup

BLE Pioneer Kit has the necessary hardware connections required for this lab. You can use your own hardware also.

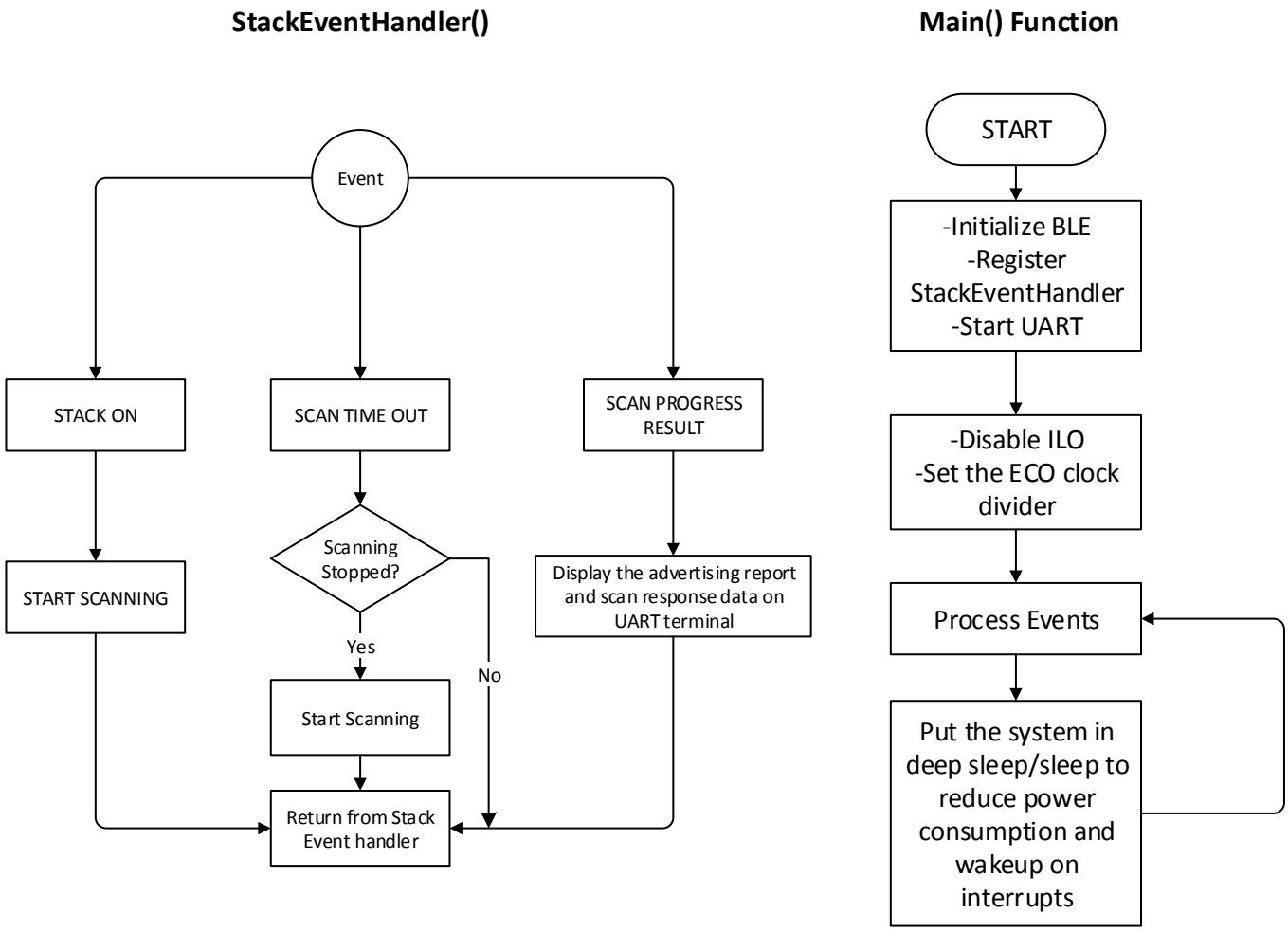
PSoC Creator Schematic

Figure 1. PSoC Creator Schematic



Operation:

Figure2. Firmware Flow



1. **main() function**: This is the central function which performs the initialization of the BLE stack, Disable ILO and set the clock divider for ECO, executes the necessary routines to process the BLE events. 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()*. CyBle_ProcessEvents processes the events received by the BLE stack and enables application layer to use them and take the appropriate action. To reduce power consumption, in between the scanning interval the system will be in deep sleep/sleep when it is not scanning.
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 and the event

CYBLE_EVT_GAPC_SCAN_PROGRESS_RESULT is received when observer receives any advertising/scan response packet from the peer devices.

3. `CYBLE_EVT_GAPC_SCAN_START_STOP:` This event occurs when scanning starts or stops. When scan time out happens in the example project it starts scanning again.

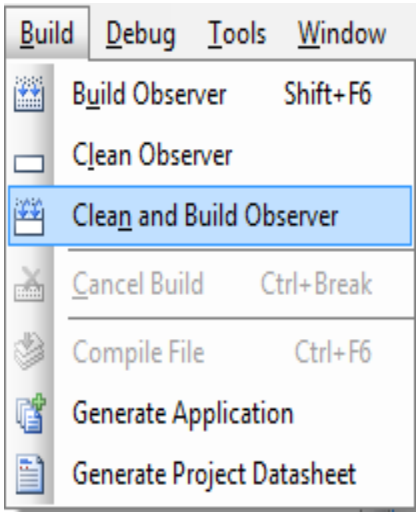
Build and Program the device

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

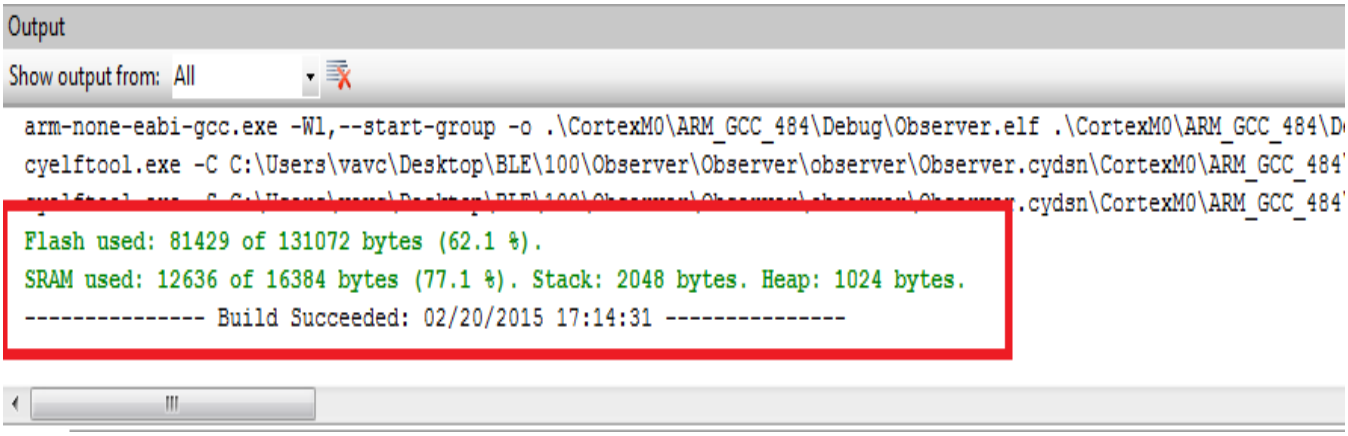
1. On PSoC Creator, select **Build** > **Clean and Build Observer**, as shown in Figure 3.

Figure 3. Build Project



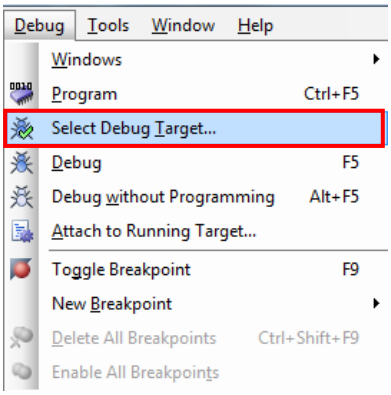
2. On successful build, total flash and SRAM usage is reported, as shown in Figure 4.

Figure 4. Build Succeeded



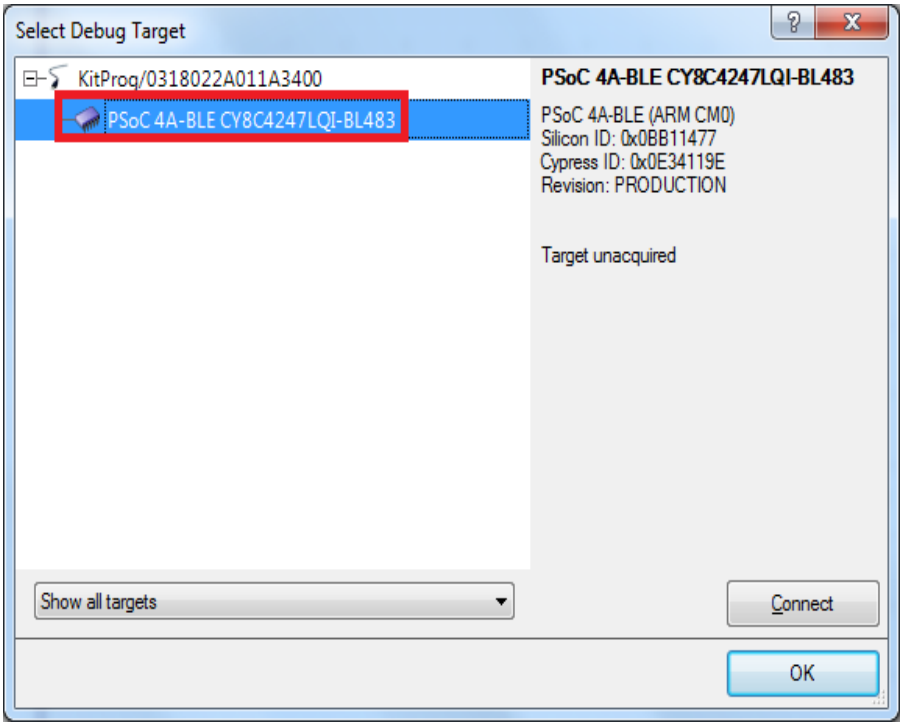
3. Select **Debug** > **Select Debug Target**, as shown in Figure5.

Figure5. Selecting Debug Target



4. In the Select Debug Target dialog box, click Port Acquire, and then click Connect, as shown in Figure6. Click OK to close the dialog box.

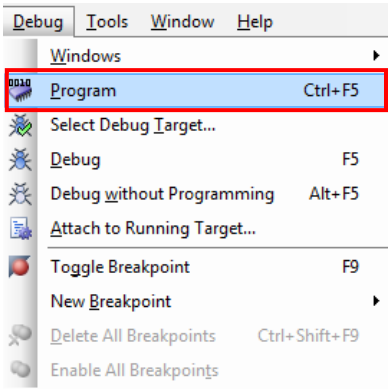
Figure6. Connecting to a Device



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.

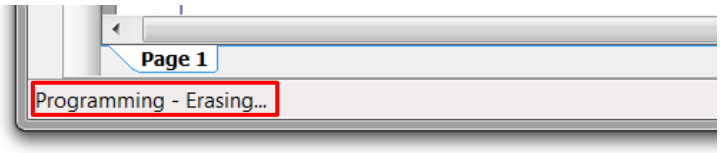
5. Select **Debug** > **Program** to program the device with the project, as shown in Figure7.

Figure7. 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.

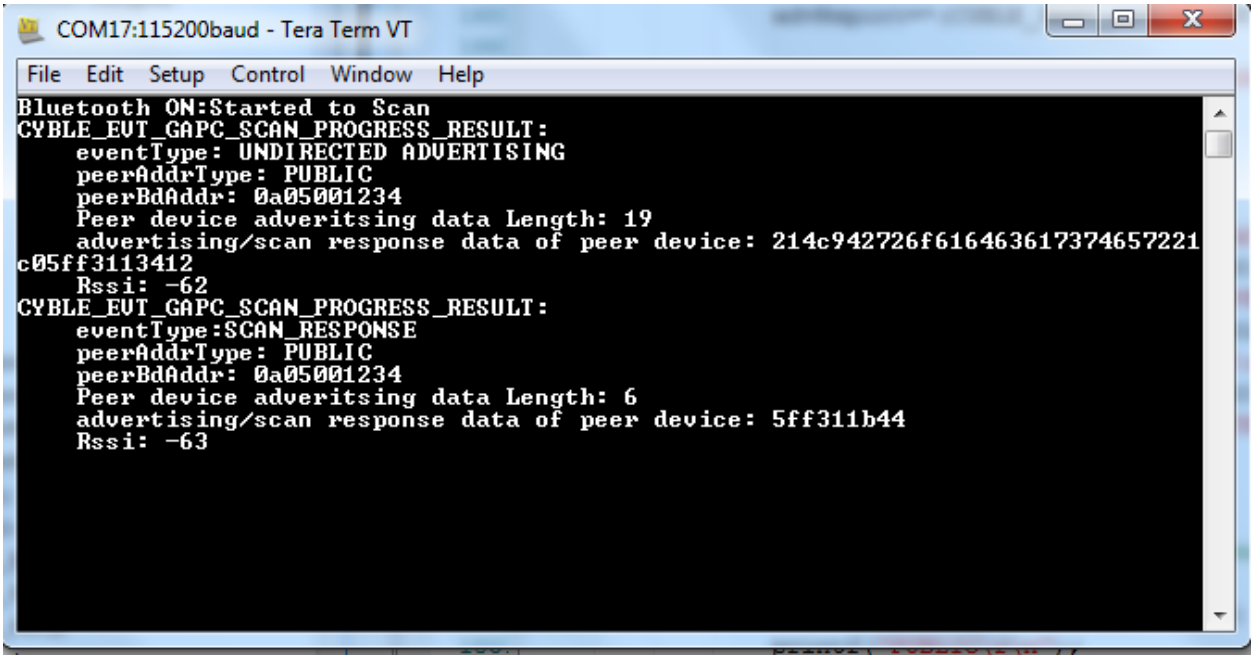
Figure 8. Programming Status



Testing:

1. Plug the BLE Pioneer Kit to your computer's USB port. Program the PSOC4 BLE device
2. Open UART terminal (like Tera Term or Hyper terminal) and use the following settings
   i. Baudrate:115200 bps
   ii. Data bits: 8
   iii. Polarity : None
   iv. Stop bits: 1bit
   v. Flow control: NO
3. Power ON any BLE device which acts as the peripheral.
4. In the UART terminal you can see received advertising and scan response packets from the peer advertising/broadcasting BLE devices.

Expected Results:



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.