PSoC 4 BLE – Power Consumption Optimization

Objective

This example demonstrates how to develop a low-power BLE system by taking advantage of the low-power modes available on the PSoC 4 BLE device and its BLE block. This project has various firmware configuration options to measure the overall system power during different device and BLE block configurations, and also acts as a toolbox to debug power issues in your design. This example could be used as a template on top of which you could develop your low-power BLE designs.

Overview

In this example the BLE Component is configured as a GAP Peripheral and a GATT Heart Rate Server. It is used for measuring the PSoC 4 BLE device power, at the BLE connection interval of 1-second (and can easily be modified for other connection intervals as well). Apart from being a reference project for measuring BLE connection interval power consumption, this example also provides firmware configuration options to:

1. Measure PSoC 4 BLE device current when in the Deep-Sleep mode
2. Measure PSoC 4 BLE device current at different operating frequencies when in the Sleep mode
3. Show the amount time for which the device is in the Sleep and Deep0Sleep modes during a BLE connection event by toggling GPIOs
4. Know when the BLE RF interface is in Tx and Rx mode by toggling GPIOs

Refer to AN92584 for a detailed description of PSoC 4 BLE power modes, current consumption details and battery life estimator tool

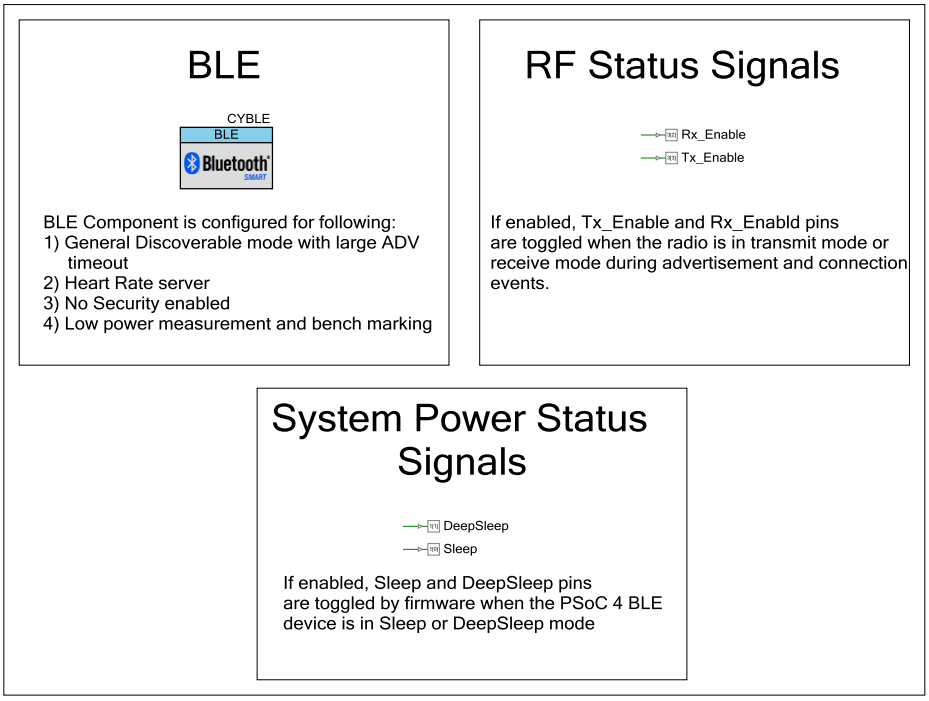
Requirements

Design Tool: PSoC Creator 3.1 SP1 with built-in GCC 4.8.4, CySmart 1.0

Associated Devices & Hardware: All PSoC 4 BLE devices & CY8CKIT-042-BLE Bluetooth® Low Energy (BLE) Pioneer Kit

PSoC Creator Schematic

Figure 1: PSoC Creator Schematic

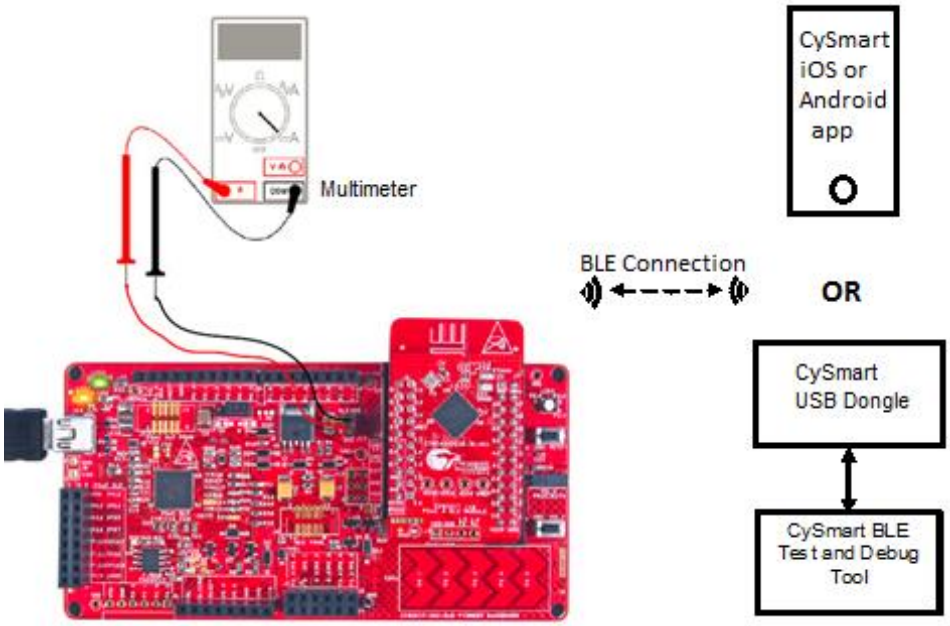


Hardware Setup

The BLE Pioneer kit programmed with this example project acts as a BLE GAP Peripheral, while the CySmart USB Dongle (included with the dev kit) acts as the BLE GAP Central. Connect a multimeter across the terminals of jumper J15 on the BLE Pioneer Baseboard to measure the isolated current consumed by the PSoC 4 BLE device (see the image below). Use the

CySmart USB Dongle along with the CySmart tool (for Windows® PC) to setup the BLE link for the required connection interval, or use an iOS/Android app if you'd like to measure the power for 1-second BLE connection interval.

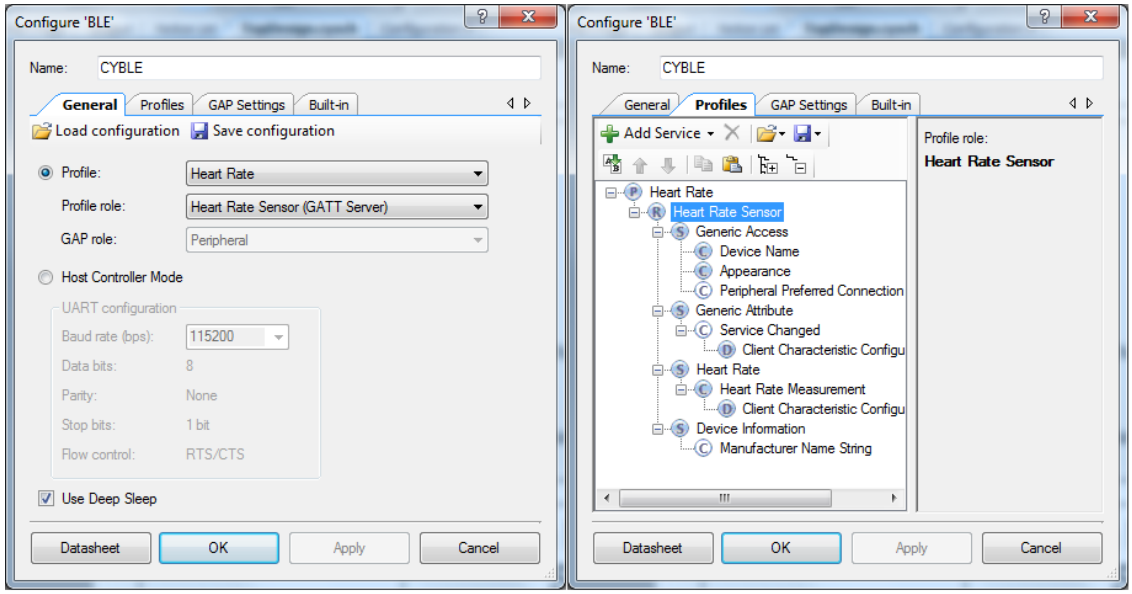
Figure 2: Hardware Setup

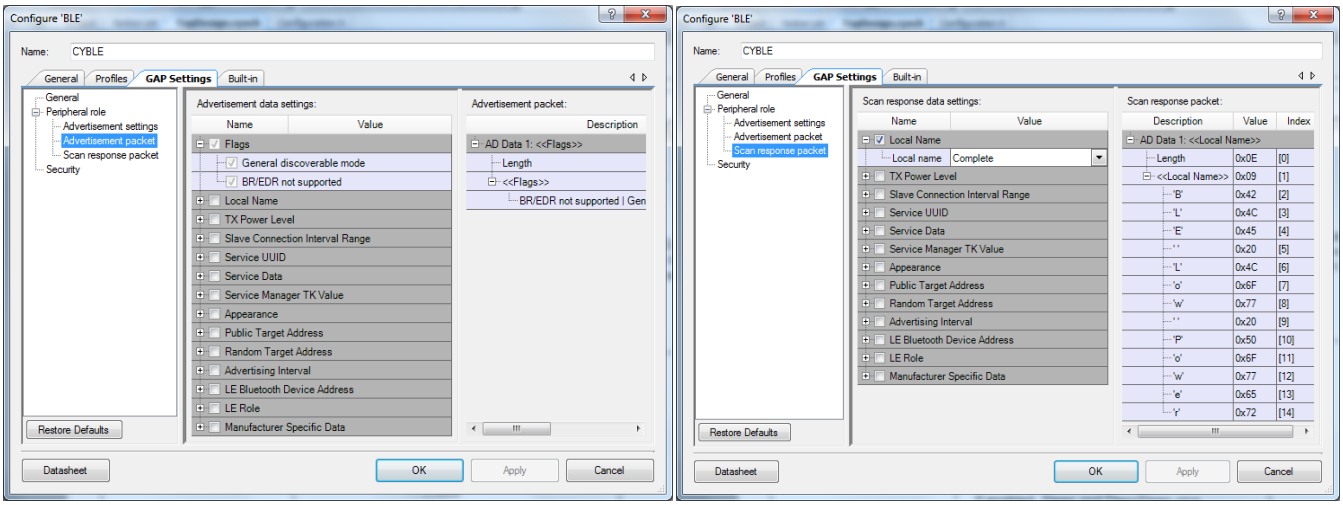


Firmware:

The BLE Component configured as a GAP Peripheral and a GATT Heart Rate Server is used as the base firmware to demonstrate power consumption. The firmware is modular and can be used as a starting point for developing end applications using PSoC 4 BLE.

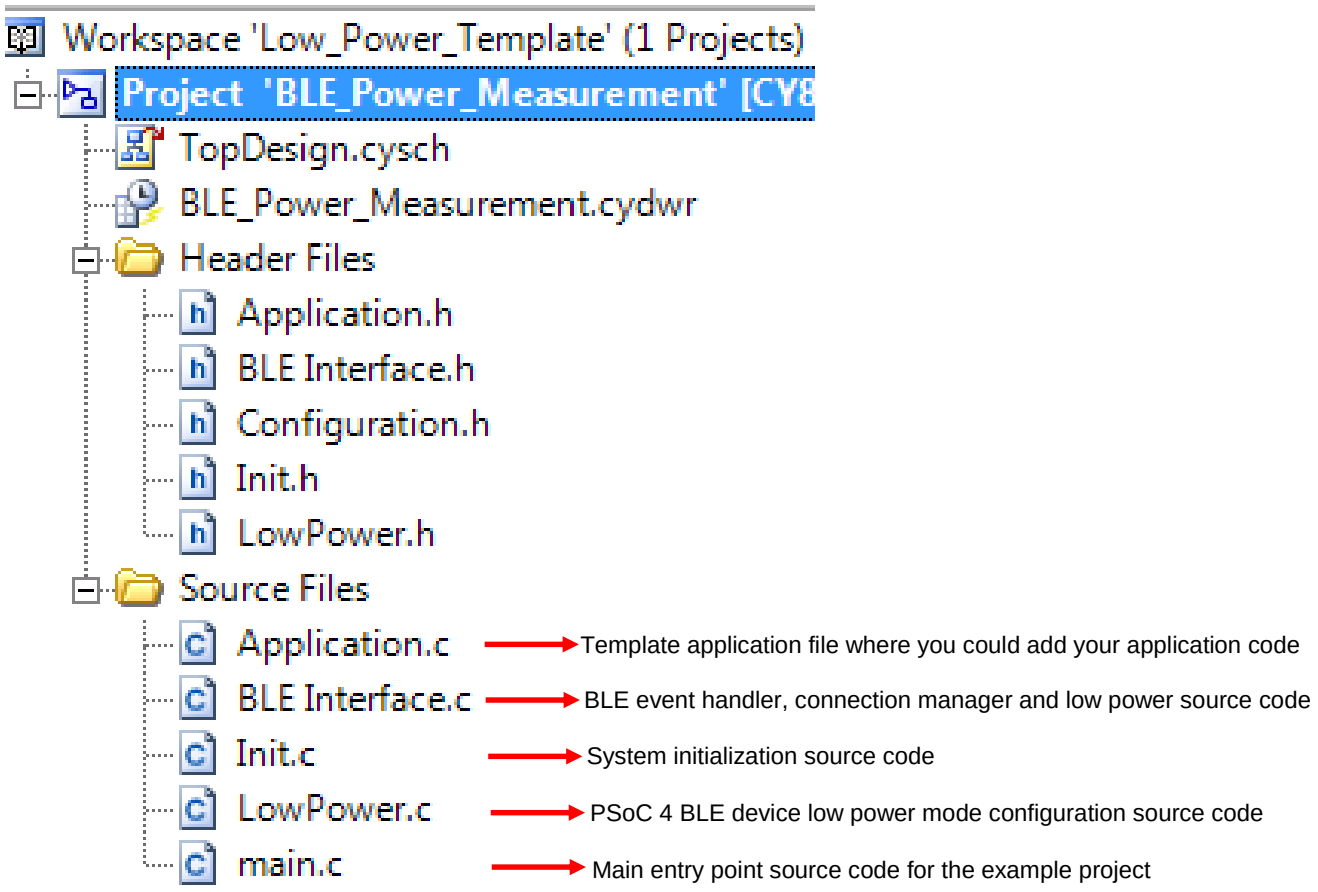
Figure 3: BLE Component Configuration





The folder structure of the source code and a brief description of each of the source files are shown below:

Figure 4: PSoC Creator Workspace Explorer



The firmware provides multiple configuration flags to test and debug different device power modes and their corresponding current consumption. All the firmware configuration flags are located in Configuration.h file and a short description of each of the configuration flags are:

Figure 5: PSoC Creator Firmware Snippet

```
/*****************************************************
 *                  Configuration flags
*****************************************************/
#define ENABLE_900_XTAL_STARTUP     1  /* Required to configure slave clock accuracy to 50ppm. Don't disable this */
#define CON_PARAM_UPDATE            1  /* When enabled, updated BLE connection interval to 1 second */
#define SEND_NOTIFICATIONS          0  /* Enables firmware to send HRM notifications when CCCD is enabled */
#define NOTIF_INTERVAL_FOUR_SEC     0  /* Sends HRM notification once in 4 connection intervals. If connection interval
                                        * is 1 sec, then the firmware sends notifications every 4 seconds */
#define DEBUG_ENABLE                1  /* Enables GPIO toggling on different power modes. Poor man's power profiler */
#define TX_RX_GPIO_ENABLE           1  /* When enabled, depicts the state of radio Tx and Rx on port 3_2 and port 3_3
                                        * respectively */
#define DEEPSLEEP_ONLY              0  /* Enable this option to measure the power consumed by PSoC 4 BLE device when in
                                        * DeepSleep mode */
#define SLEEP_ONLY                  0  /* Enable this option to measure the power consumed by PSoC 4 BLE device when in
                                        * Sleep mode at operating frequency set by SLEEP_OPERATING_FREQUENCY (see below)*/

#if SLEEP_ONLY
#define SLEEP_OPERATING_FREQUENCY   16 /* IMO frequency in MHz for which PSoC 4 BLE device sleep current is to be measured */
#endif

#if CON_PARAM_UPDATE
    #define NOTIFICATION_OFFSET     10 /* When to start sending notification after CCCD is enabled */
#else
    #define NOTIFICATION_OFFSET     1
#endif /* End of #if CON_PARAM_UPDATE */

#endif /* End of #if !defined(CONFIGURATION_H) */
```

The firmware continuously advertises with an advertisement interval range of 20 to 30ms while waiting for a BLE GAP Central device to initiate a connection. Once connected, it maintains the BLE link with the GAP Central device and waits for Heart Rate Notifications to be enabled by the Heart Rate Collector on the GAP Central device. Once the Heart Rate Notifications are enabled, depending on the firmware configuration flag settings, one or more of the following will happen:

- If the CON_PARAM_UPDATE configuration flag is set, then the firmware sends a L2CAP connection parameter update procedure to the GAP Central device requesting it to override the default BLE connection interval with a 1-second connection interval. This setting is useful to measure the idle mode current consumption for a 1-second BLE connection interval.
- If the SEND_NOTIFICATION flag is set, then the firmware sends one Heart Rate Notification packet every connection interval so that current consumption of the PSoC 4 BLE device while transferring a non-zero payload can be measured.
- If the NOTIF_INTERVAL_FOUR_SEC is set, once the Heart Rate Notifications are enabled by the Heart Rate Collector, the firmware sends one notification packet for four connection interval periods. This setting, in conjunction with the CON_PARAM_UPDATE flag enables you to measure 4-second data transfer power consumption of the PSoC 4 BLE device.

Refer to AN92584 for a list of tips and tricks for achieving low power consumption in PSoC 4 BLE and PRoC BLE devices.

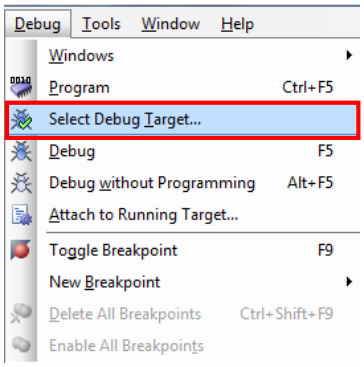
Build and Program

This section shows how to build the project and program the PSoC 4 BLE device on the BLE Pioneer Baseboard.

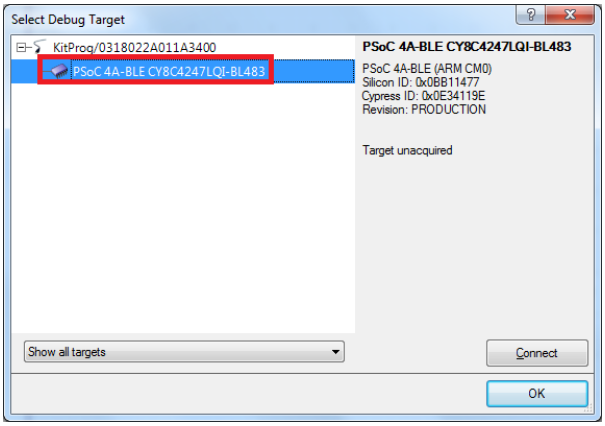
1. Open PSoC creator 3.1 SP1. Go to **File -> Open -> Project / Workspace**. Browse for the folder containing the project files and select **BLE_Power_Measurement.cyprj**.
2. Go to **Build -> Build BLE_Power_Measurement.**
3. On a successful build, the total flash and SRAM usage is reported as shown below
4. Select **Debug** > **Select Debug Target**, as shown below.

Figure 6: Build and Program

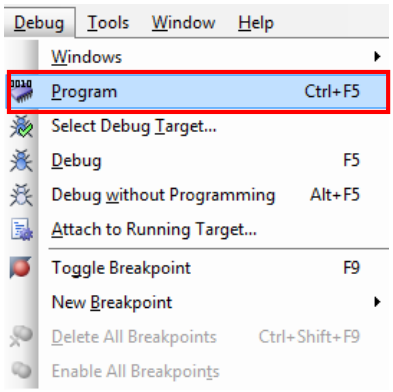
```
Flash used: 73159 of 131072 bytes (55.8 %).
SRAM used: 11492 of 16384 bytes (70.1 %). Stack: 2048 bytes. Heap: 0 bytes.
--------------- Build Succeeded: 03/24/2015 02:39:03 ---------------
```



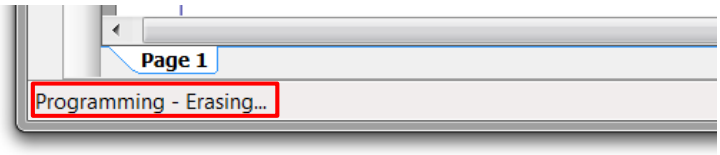
5. In the **Select Debug Target** dialog box, click **Port Acquire**, and then click **Connect** as shown below. Click **OK** to close the dialog box.



6. Select **Debug** > **Program** to program the device with the project, as shown below.



You can view the programming status on the PSoC Creator status bar (lower-left corner of the window), as shown below.

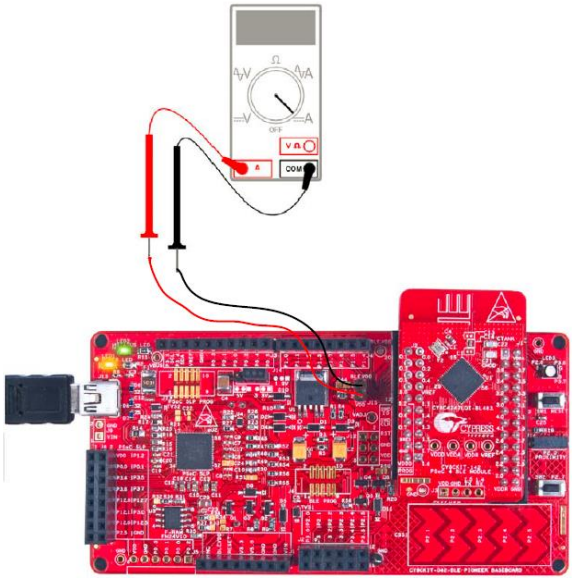


PSoC 4 BLE – Power Consumption Optimization

Test setup

1. Program the hex file generated after building this project onto BLE Pioneer Baseboard.
2. Connect a Agilent 6.5 digit multimeter or equivalent device across jumper J15 on BLE Pioneer Baseboard
3. Set the multimeter aperture to 1 second, range to >25mA (auto-range won't work), and set continuous average mode on
4. Use an iPhone or Android device as the BLE Central device for power measurement as they have an accurate clock (<50ppm). If you use CySmart dongle, the current number will be higher due to lower clock accuracy (150ppm)

Current Measurement when Powered from USB Port

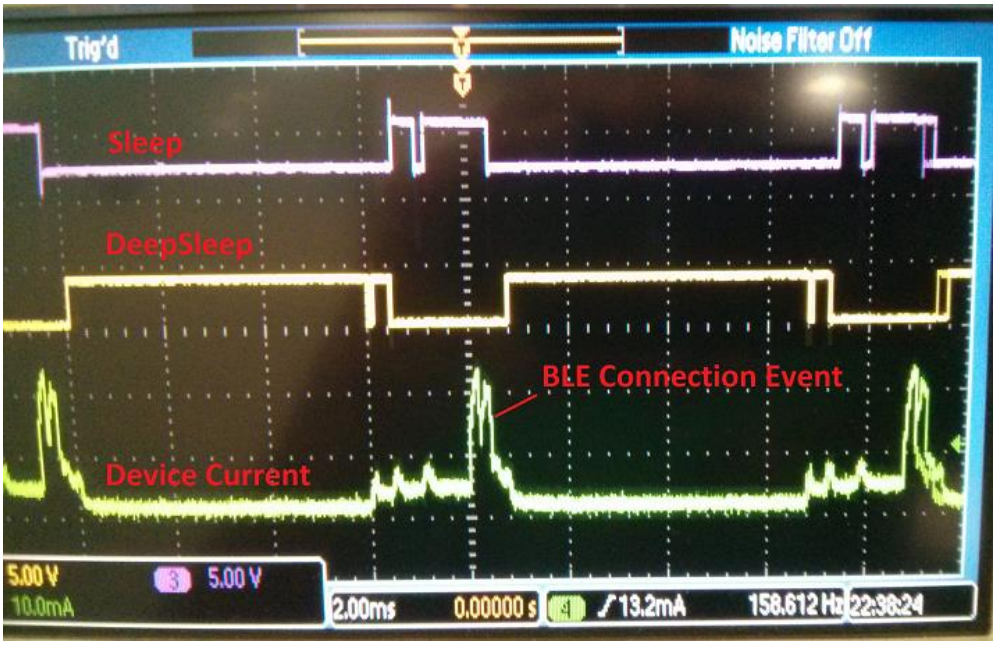


Test procedure

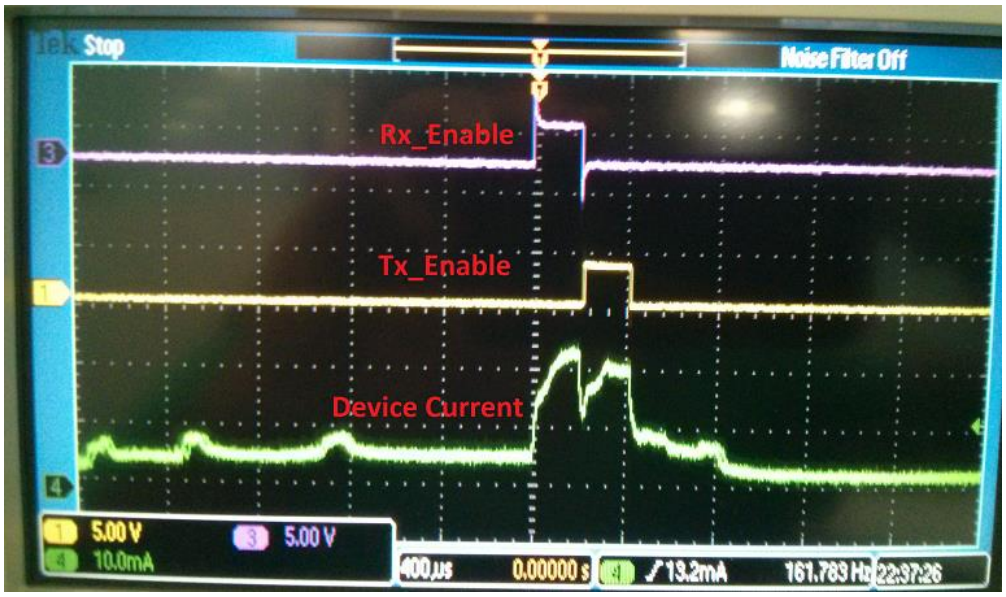
1. Establish a connection between iPhone/Android device and the BLE Pioneer kit
2. Enable Heart Rate Notifications, this updates the connection interval to 1 second, required for making current measurement
3. Measure the current now and you should be able to measure a value of <18.9uA for 1 second connection interval

Current and debug plots

The following figure shows the PSoC 4 BLE device power plots and the state of Sleep and Deep-Sleep debug GPIOs (when DEBUG_ENABLE flag is set in firmware) during an active BLE connection interval.



Similarly, if TX_RX_GPIO_ENABLE flag is set in firmware, the state of Tx_Enable and Rx_Enable debug GPIOs with respect to actual BLE RF event is as below:



Related Documents

The table below lists all relevant application notes, code examples, knowledge base articles, device datasheets, and Component / user module datasheets.

Related Documents

Document	Title	Comment
AN91267	Getting Started with PSoC® 4 BLE	A guide for beginners on PSoC 4 BLE
AN92584	Designing for Low Power and Estimating Battery Life for BLE Applications	BLE low power application design guide
001-90479	Programmable System-on-Chip (PSoC®)	PSoC® 4: PSoC 4XX7_BLE Family Datasheet