LAB 3 REPORT
IOT SENSOR-BASED SYSTEM DESIGN

Director: Dr. Ahmed Rafaey Hussein

Group Members: Mohammadamin Saburruhmonfared & Chaoao Shi

June 17, 2016

OBJECTIVES

The main purpose of this lab is to create a heart rate sensor device which can report the measured heart rate value to a BLE enabled device. In order to simplify the hardware platform, the input heart rate signal is an analog signal on the PSoC 4 BLE device. The objectives of Lab 3 include:

1. Measure simulated heart rate using the Programmable Analog Blocks
2. Implement a Heart Rate Profile and send the data over BLE
3. Optimize the design for low power consumption using Sleep, Deep-Sleep and Hibernate modes

PROCEDURE

1. Block diagram

Set up the project in PSoC Creator so that it follows the block diagram shown in Figure 1.

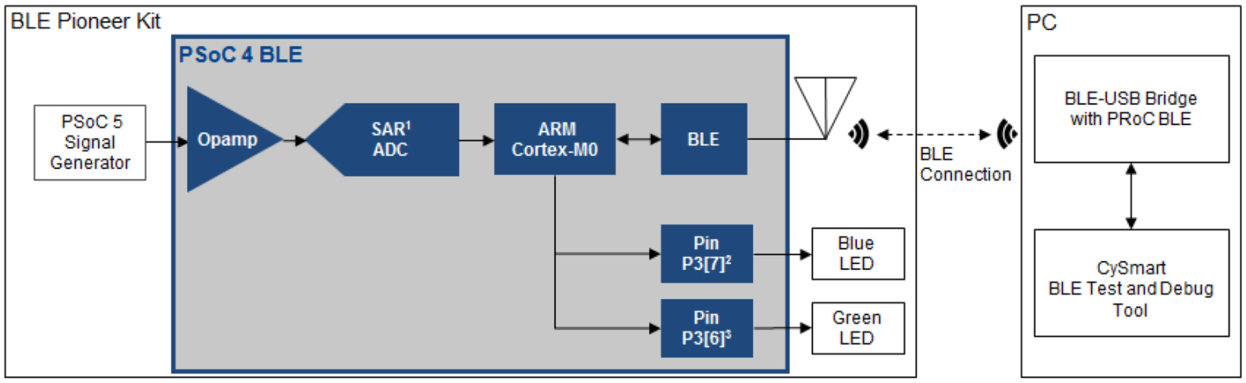
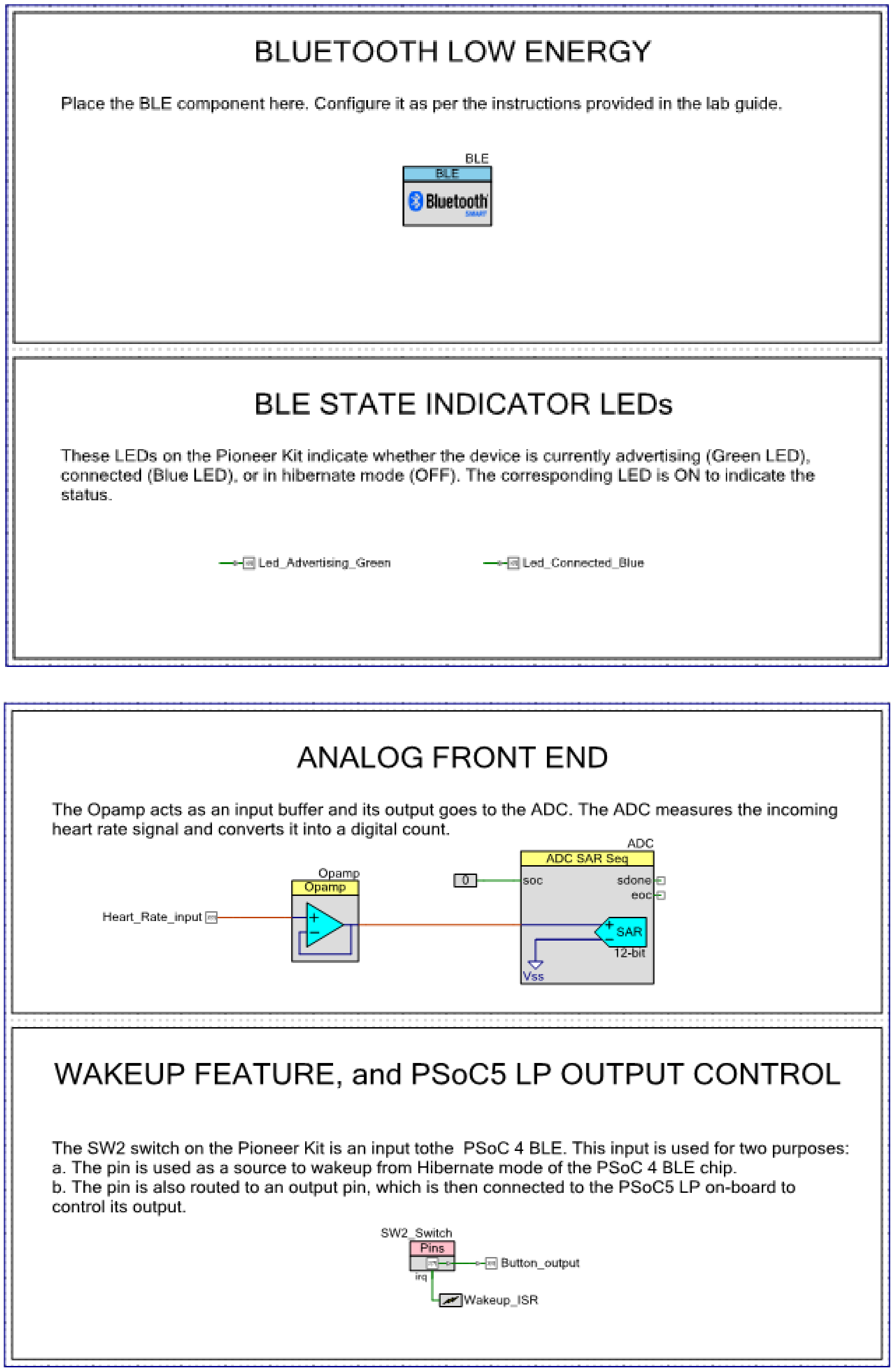


Figure 1: Block Diagram for Lab 2

2. Schematic

Open the BLE Lab 3 template provided by Cypress, and open TopDesign.cysch. In the Bluetooth Low Energy sheet of the schematic, place the BLE component, and drag the Opamp Component and Sequencing SAR ADC to the Analog Front End sheet, as shown in Figure 2.


BLUETOOTH LOW ENERGY

Place the BLE component here. Configure it as per the instructions provided in the lab guide.

BLE STATE INDICATOR LEDs

These LEDs on the Pioneer Kit indicate whether the device is currently advertising (Green LED), connected (Blue LED), or in hibernate mode (OFF). The corresponding LED is ON to indicate the status.

ANALOG FRONT END

The Opamp acts as an input buffer and its output goes to the ADC. The ADC measures the incoming heart rate signal and converts it into a digital count.

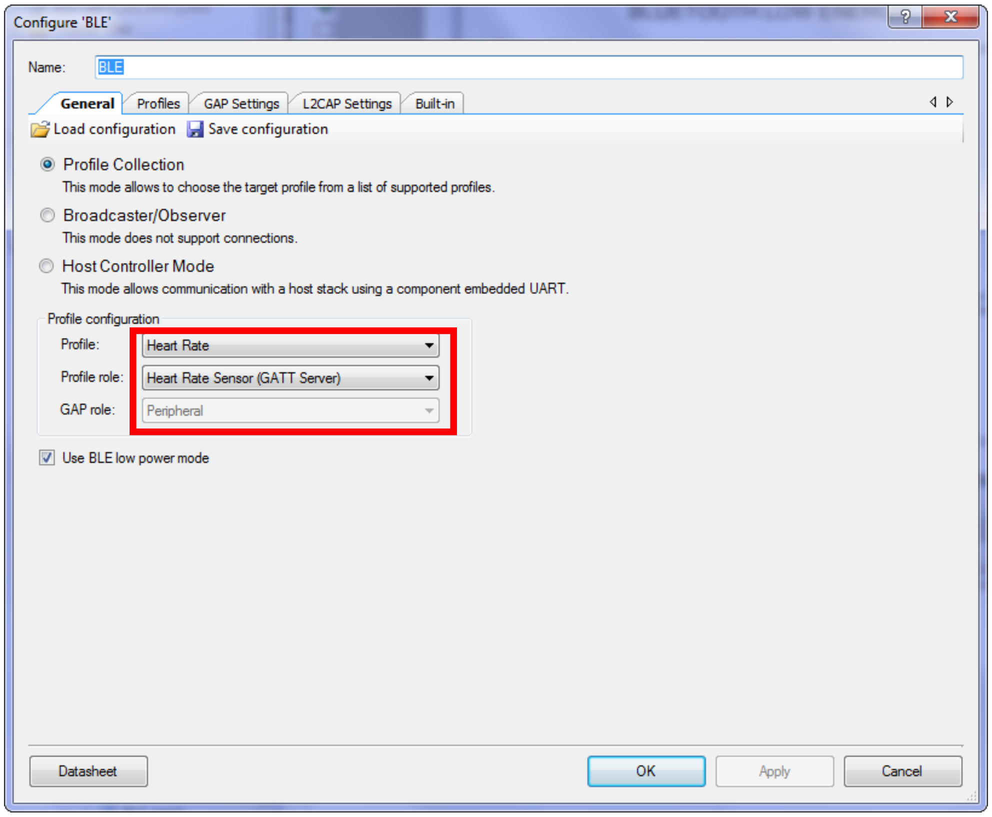
WAKEUP FEATURE, and PSoC5 LP OUTPUT CONTROL

The SW2 switch on the Pioneer Kit is an input tothe PSoC 4 BLE. This input is used for two purposes:
a. The pin is used as a source to wakeup from Hibernate mode of the PSoC 4 BLE chip.
b. The pin is also routed to an output pin, which is then connected to the PSoC5 LP on-board to control its output.

Figure 2: Schematic

3. BLE configuration – General Tab

Set the profile to Heart Rate and the Profile role to Heart Rate Sensor (GATT server).


Configure 'BLE'

Name: BLE

General | Profiles | GAP Settings | L2CAP Settings | Built-in

Load configuration | Save configuration

● Profile Collection
This mode allows to choose the target profile from a list of supported profiles.

○ Broadcaster/Observer
This mode does not support connections.

○ Host Controller Mode
This mode allows communication with a host stack using a component embedded UART.

Profile configuration
Profile: Heart Rate
Profile role: Heart Rate Sensor (GATT Server)
GAP role: Peripheral

☑ Use BLE low power mode

Datasheet | OK | Apply | Cancel

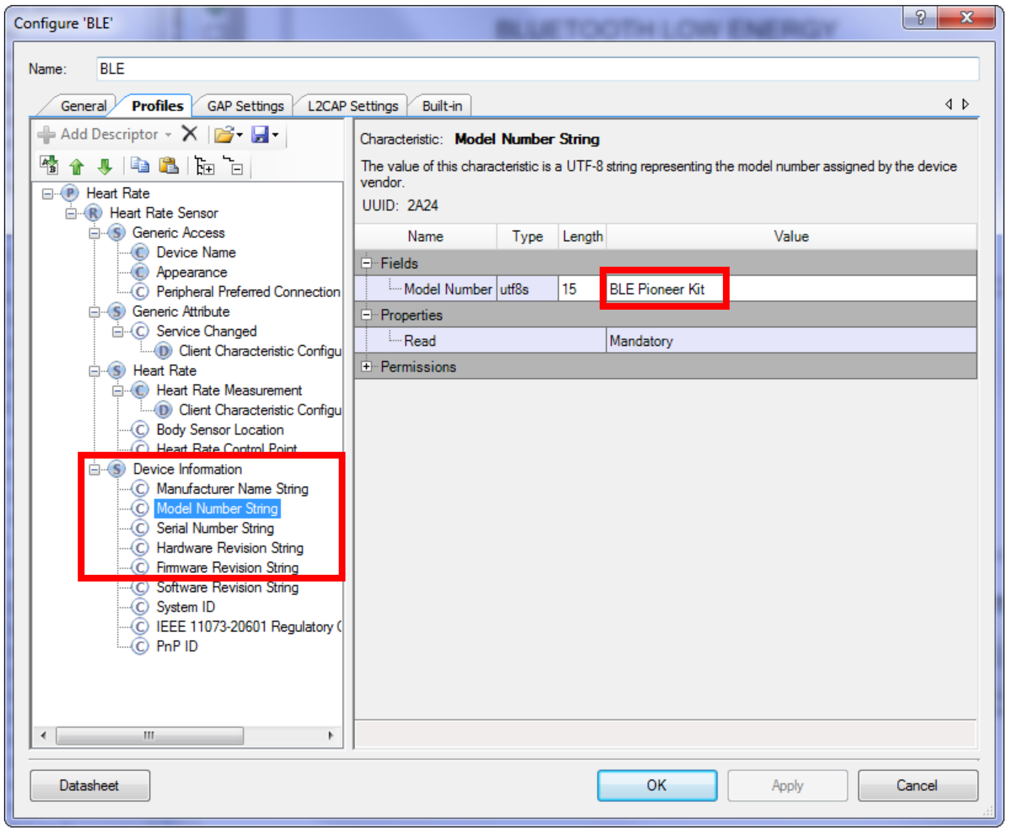
Figure 3: BLE Component Configuration – General Tab

4. BLE configuration – Profiles Tab

According to Table 1, assign the values to the Characteristics of the DIS. These values can be read on the GATT Client BLE device.

Table 1: Device Information Service Characteristics

Characteristic	Field	Value
Manufacturer Name String	Manufacturer Name	Cypress Semiconductor
Model Number String	Model Number	BLE Pioneer Kit
Serial Number String	Serial Number	1
Hardware Revision String	Hardware Revision	**
Firmware Revision String	Firmware Revision	1.0



Name: BLE

- Heart Rate
  - Heart Rate Sensor
    - Generic Access
      - Device Name
      - Appearance
      - Peripheral Preferred Connection
    - Generic Attribute
      - Service Changed
        - Client Characteristic Configu
    - Heart Rate
      - Heart Rate Measurement
        - Client Characteristic Configu
      - Body Sensor Location
      - Heart Rate Control Point
    - Device Information
      - Manufacturer Name String
      - Model Number String
      - Serial Number String
      - Hardware Revision String
      - Firmware Revision String
      - Software Revision String
      - System ID
      - IEEE 11073-20601 Regulatory (
      - PnP ID

Characteristic: **Model Number String**

The value of this characteristic is a UTF-8 string representing the model number assigned by the device vendor.

UUID: 2A24

Name	Type	Length	Value
Fields			
Model Number	utf8s	15	BLE Pioneer Kit
Properties			
Read			Mandatory
Permissions			

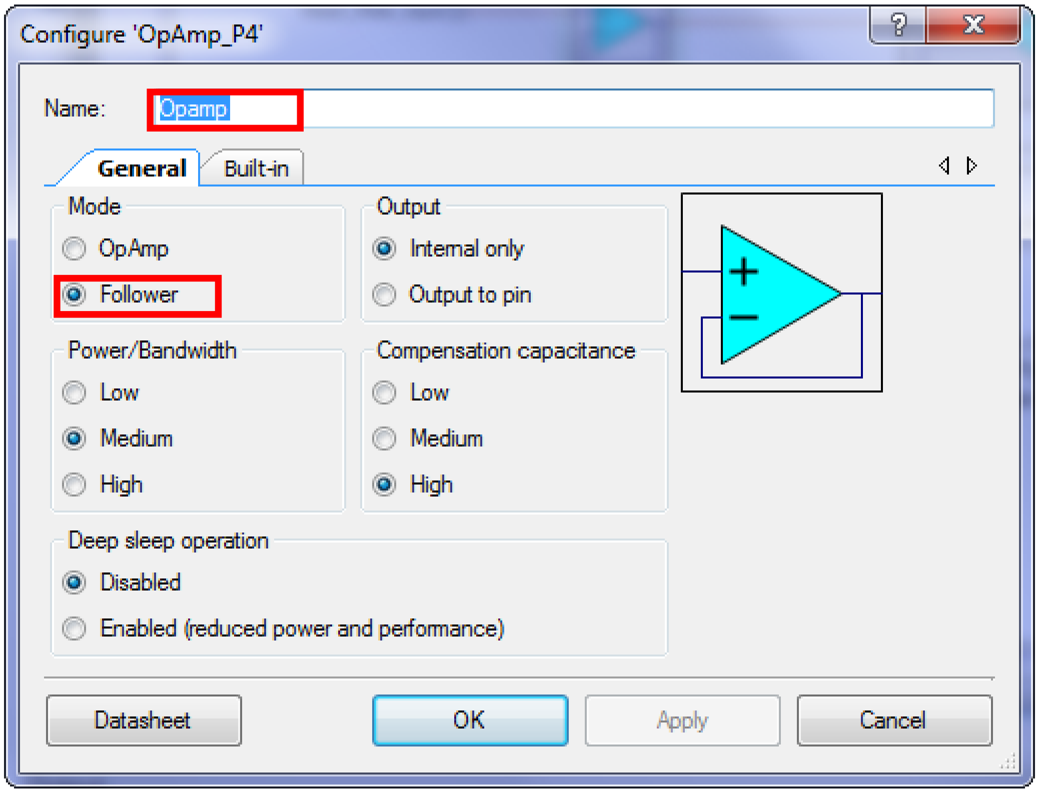
Figure 4: GAP Setting Tab – Profiles

5. BLE configuration – Others

The BLE component needs other configurations such as General, Peripheral Role and Security in GAP Settings Tag. Since these settings are the same to settings in Lab 2, I decided not to include this part in the Lab 3 report.

6. Opamp component

Place the Opamp Component and set the Mode to Follower.


Configure 'OpAmp_P4'

Name: Opamp

General | Built-in

Mode
- OpAmp
- [✓ Follower]

Output
- [✓ Internal only]
- Output to pin

Power/Bandwidth
- Low
- [✓ Medium]
- High

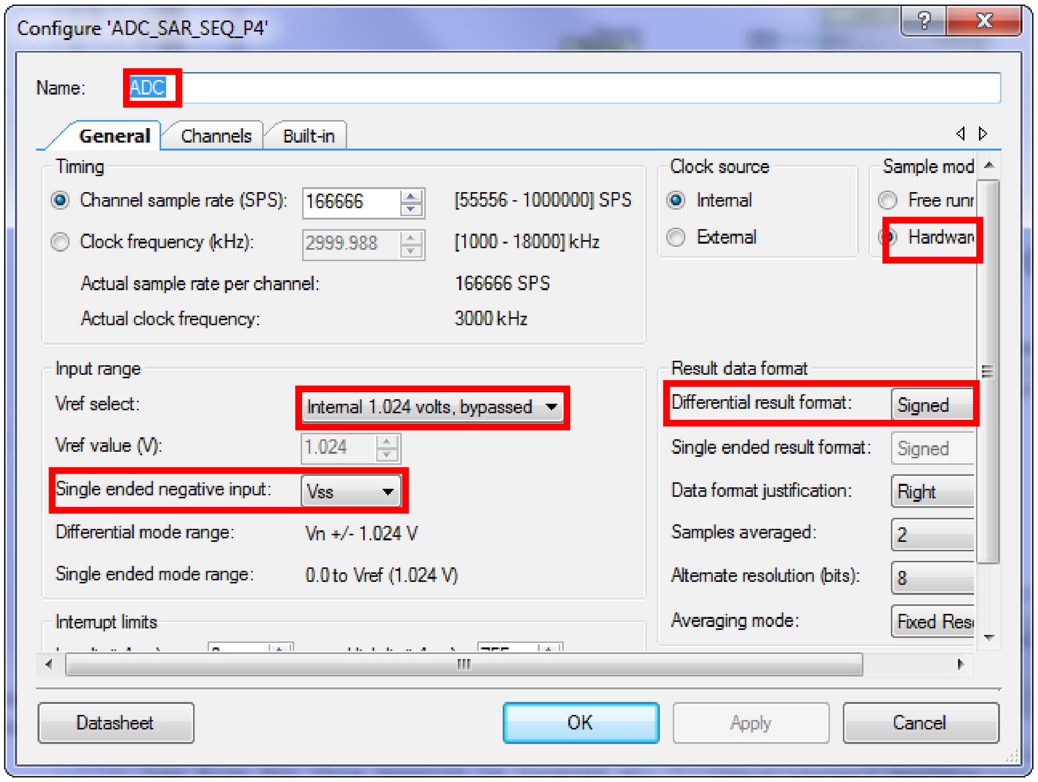
Compensation capacitance
- Low
- Medium
- [✓ High]

Deep sleep operation
- [✓ Disabled]
- Enabled (reduced power and performance)

Figure 5: Opamp Component Configuration

7. ADC component

Place the Sequencing SAR ADC Component in the Analog Front End Sheet, and set it as shown in Figure 6 and Figure 7.


Configure 'ADC_SAR_SEQ_P4'

Name: ADC

General | Channels | Built-in

Timing

- (●) Channel sample rate (SPS): 166666 [55556 - 1000000] SPS
- ( ) Clock frequency (kHz): 2999.988 [1000 - 18000] kHz
- Actual sample rate per channel: 166666 SPS
- Actual clock frequency: 3000 kHz

Clock source

- (●) Internal
- ( ) External

Sample mod

- ( ) Free runr
- (●) Hardwar

Input range

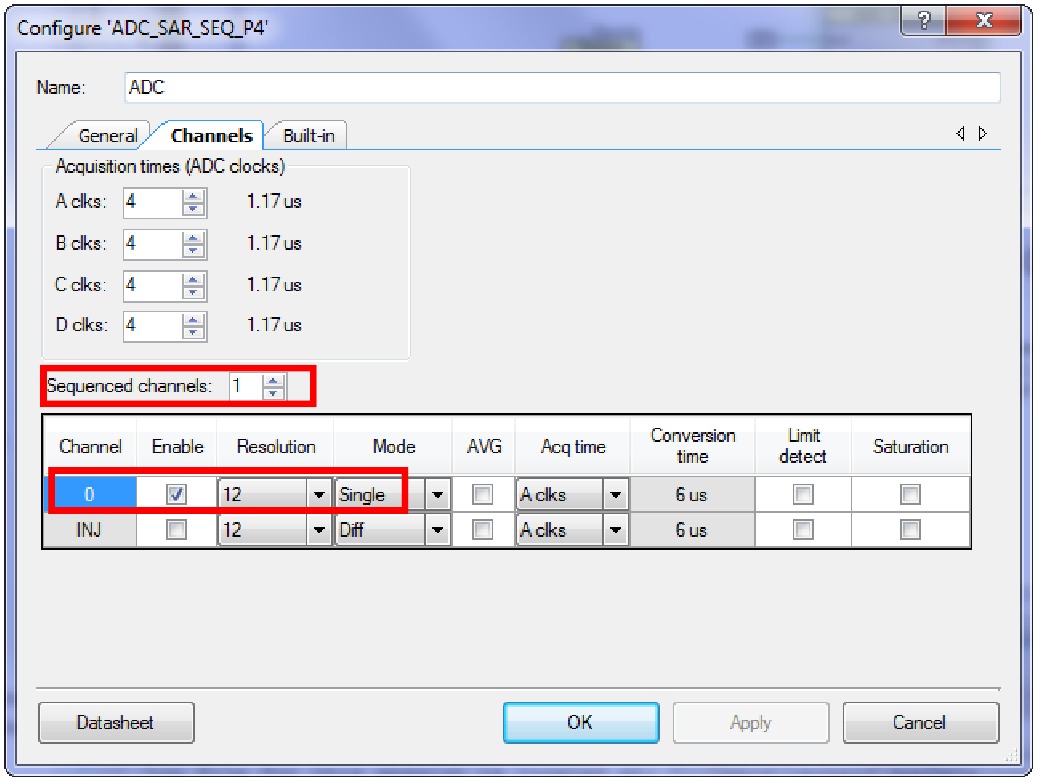
Vref select:	Internal 1.024 volts, bypassed
Vref value (V):	1.024
Single ended negative input:	Vss
Differential mode range:	Vn +/- 1.024 V
Single ended mode range:	0.0 to Vref (1.024 V)

Result data format

Differential result format:	Signed
Single ended result format:	Signed
Data format justification:	Right
Samples averaged:	2
Alternate resolution (bits):	8
Averaging mode:	Fixed Res

Interrupt limits

Figure 6: ADC General Settings


Configure 'ADC_SAR_SEQ_P4'

Name: ADC

General | Channels | Built-in

Acquisition times (ADC clocks)

A clks:	4	1.17 us
B clks:	4	1.17 us
C clks:	4	1.17 us
D clks:	4	1.17 us

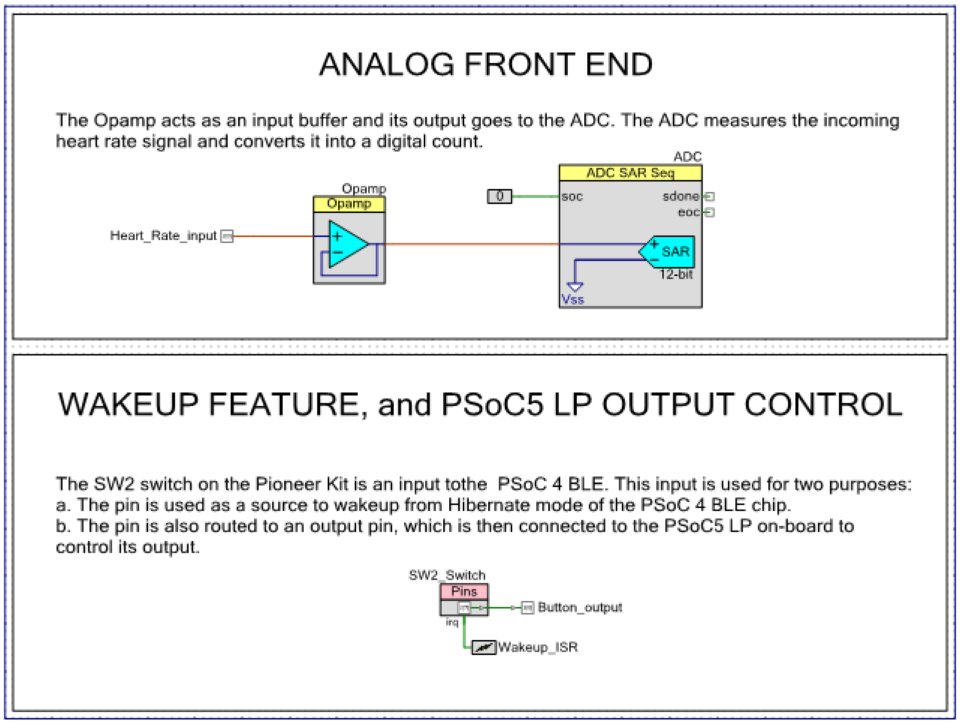
Sequenced channels: 1

Channel	Enable	Resolution	Mode	AVG	Acq time	Conversion time	Limit detect	Saturation
0	✓	12	Single	☐	A clks	6 us	☐	☐
INJ	☐	12	Diff	☐	A clks	6 us	☐	☐

Figure 7: ADC Channel Settings

8. Complete the schematic

Add the Logic Low ‘0’ Component to the schematic editor and connect its output to the soc input of the ADC Component. Connect the Heart_Rate_input pin terminal to the + input of the Opamp. Connect the output of the Opamp to the + input of the ADC. See Figure 8.



ANALOG FRONT END

The Opamp acts as an input buffer and its output goes to the ADC. The ADC measures the incoming heart rate signal and converts it into a digital count.

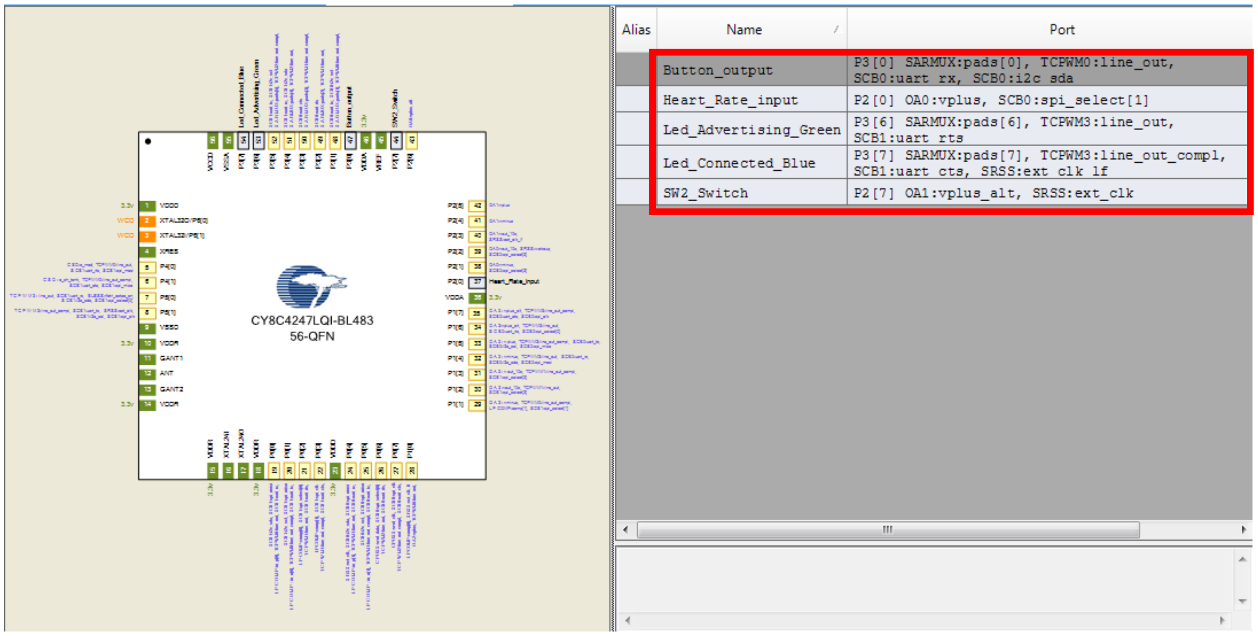
WAKEUP FEATURE, and PSoC5 LP OUTPUT CONTROL

The SW2 switch on the Pioneer Kit is an input tothe PSoC 4 BLE. This input is used for two purposes:
a. The pin is used as a source to wakeup from Hibernate mode of the PSoC 4 BLE chip.
b. The pin is also routed to an output pin, which is then connected to the PSoC5 LP on-board to control its output.

Figure 8: Analog Front End Sheet of the Schematic

9. DWR configuration

Now we’re done with configuring the schematics of the BLE and the ADC component. We’re going to configure the DWR file, which is a Design Wide Resources. The results are shown in Figure 9.

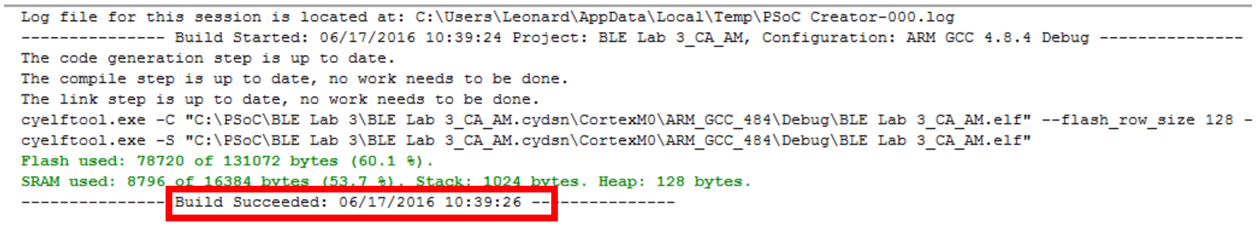


Alias	Name	Port
	Button_output	P3[0] SARMUX:pads[0], TCPWM0:line_out, SCB0:uart rx, SCB0:i2c sda
	Heart_Rate_input	P2[0] OA0:vplus, SCB0:spi_select[1]
	Led_Advertising_Green	P3[6] SARMUX:pads[6], TCPWM3:line_out, SCB1:uart rts
	Led_Connected_Blue	P3[7] SARMUX:pads[7], TCPWM3:line_out_compl, SCB1:uart cts, SRSS:ext clk lf
	SW2_Switch	P2[7] OA1:vplus_alt, SRSS:ext_clk

Figure 9: DWR Configuration

10. Build

Then we build the application to generate source code files, which will auto-complete the API names, variables and macro. Figure 10 shows that build is successful.



```
Log file for this session is located at: C:\Users\Leonard\AppData\Local\Temp\PSoC Creator-000.log
--------------- Build Started: 06/17/2016 10:39:24 Project: BLE Lab 3_CA_AM, Configuration: ARM GCC 4.8.4 Debug ---------------
The code generation step is up to date.
The compile step is up to date, no work needs to be done.
The link step is up to date, no work needs to be done.
cyelftool.exe -C "C:\PSoC\BLE Lab 3\BLE Lab 3_CA_AM.cydsn\CortexM0\ARM_GCC_484\Debug\BLE Lab 3_CA_AM.elf" --flash_row_size 128 -
cyelftool.exe -S "C:\PSoC\BLE Lab 3\BLE Lab 3_CA_AM.cydsn\CortexM0\ARM_GCC_484\Debug\BLE Lab 3_CA_AM.elf"
Flash used: 78720 of 131072 bytes (60.1 %).
SRAM used: 8796 of 16384 bytes (53.7 %). Stack: 1024 bytes. Heap: 128 bytes.
--------------- Build Succeeded: 06/17/2016 10:39:26 ---------------
```

Figure 10: Successful Project Build

11. Program

Before connecting the kit to the PC, we should press the SW1 (Reset) switch, and at the same time plug in the kit’s USB connector to the PC. This puts the kit into the bootloader mode, when the Status LED (LED 2) starts blinking at a frequency of 1 Hz.

Then we launched the Bootloader Host Tool and set parameters in VID.

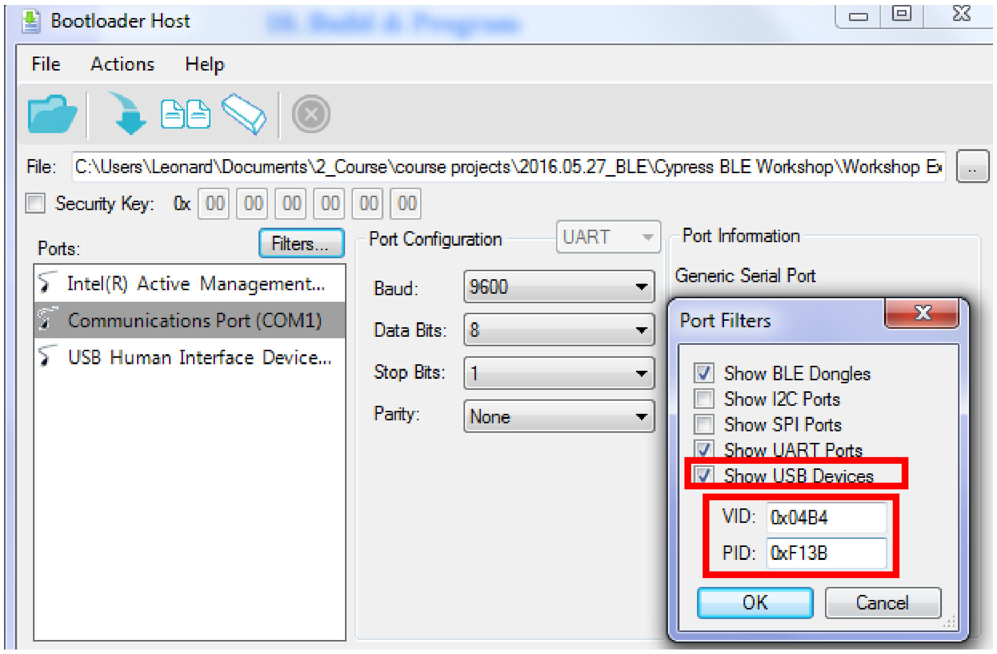
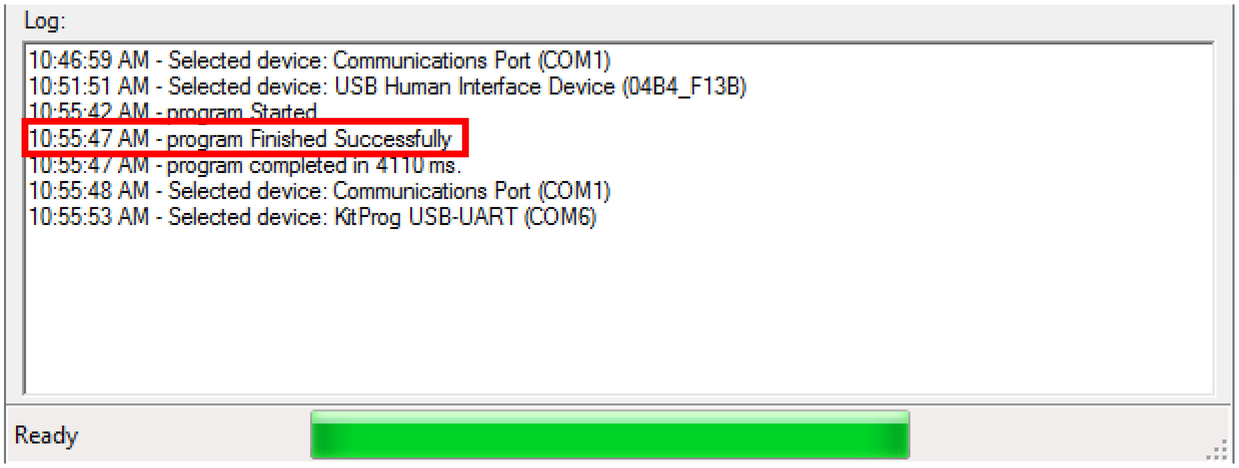


Figure 11: Adding a Device Filter in the Bootloader Host Tool

By selecting the USB Human Interface Device, we opened the bootloadable file named **KitProg_custom.cyacd**. We programmed the PSoc 5 on the kit using the Program menu item. The program log is shown in Figure 12.



```
Log:
10:46:59 AM - Selected device: Communications Port (COM1)
10:51:51 AM - Selected device: USB Human Interface Device (04B4_F13B)
10:55:42 AM - program Started
10:55:47 AM - program Finished Successfully
10:55:47 AM - program completed in 4110 ms.
10:55:48 AM - Selected device: Communications Port (COM1)
10:55:53 AM - Selected device: KitProg USB-UART (COM6)
Ready
```

Figure 12: Successful Programmed to Kit

RESULTS

1. Firmware Flow

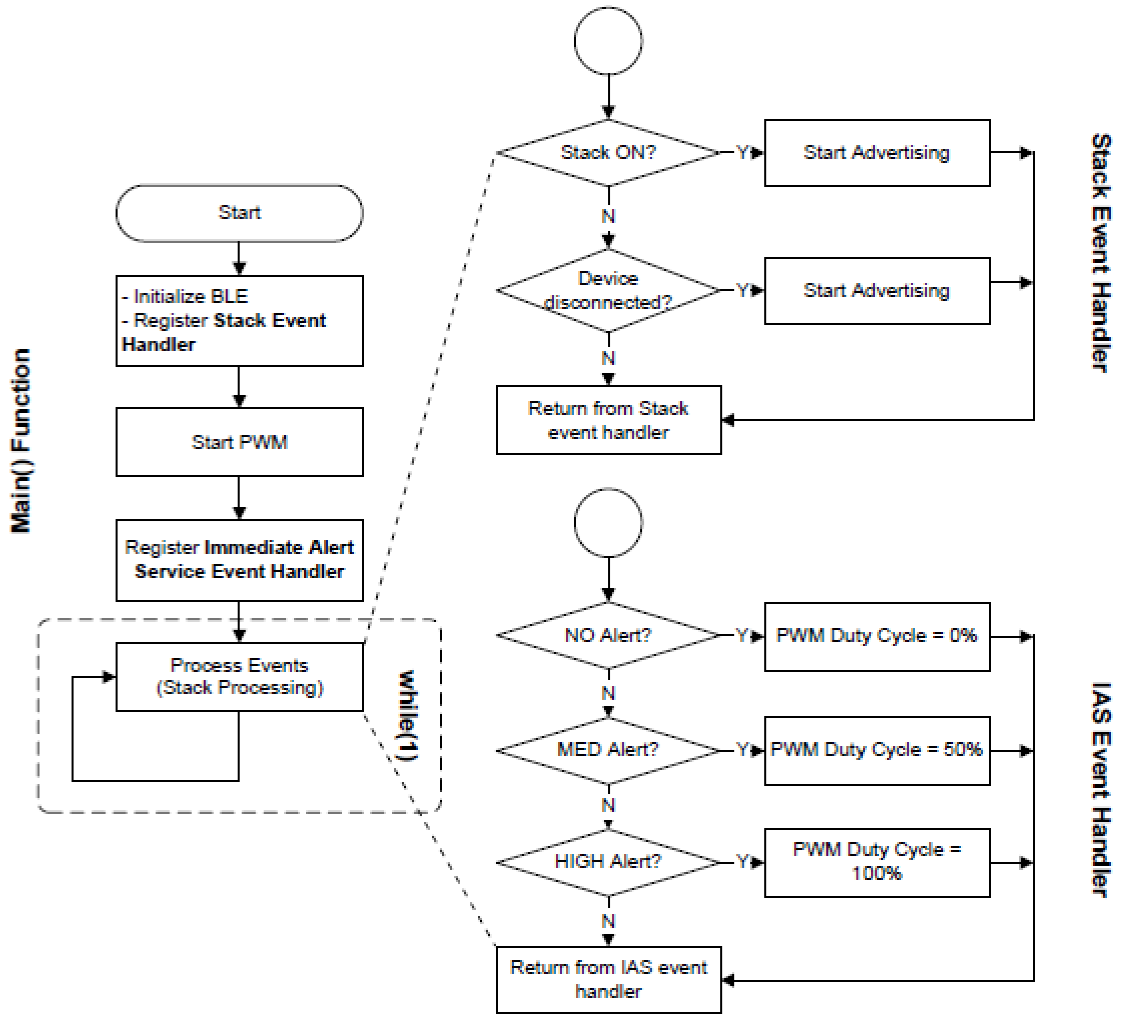
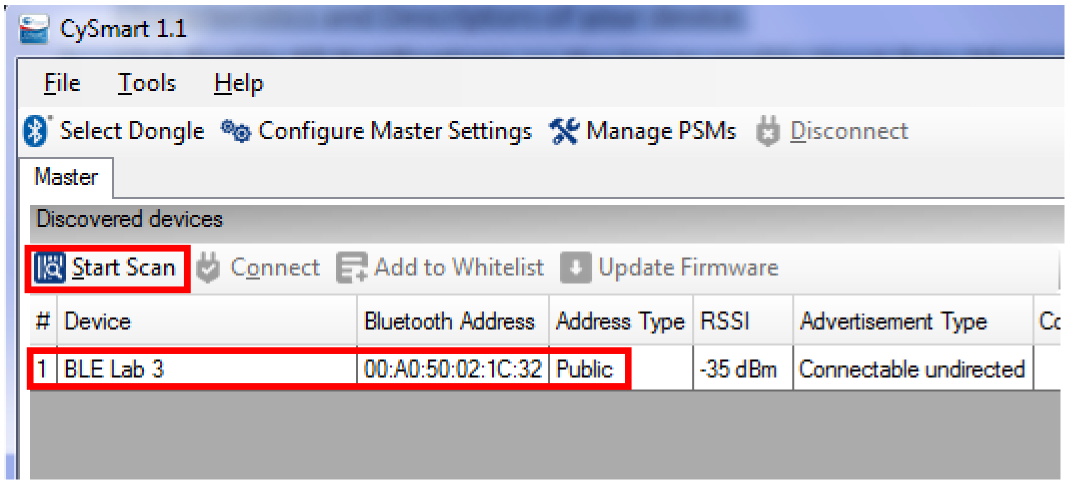


Figure 13: Firmware Flow

2. Testing: CySmart 1.1 Connection

We plugged the BLE-USB Bridge (included with the BLE Pioneer Kit) in my computer's USB port, and then launched CySmart 1.1. By connecting the Cypress BLE Dongle and using it to scan BLE device, we can find the BLE Pioneer Kit, as shown in Figure 14.

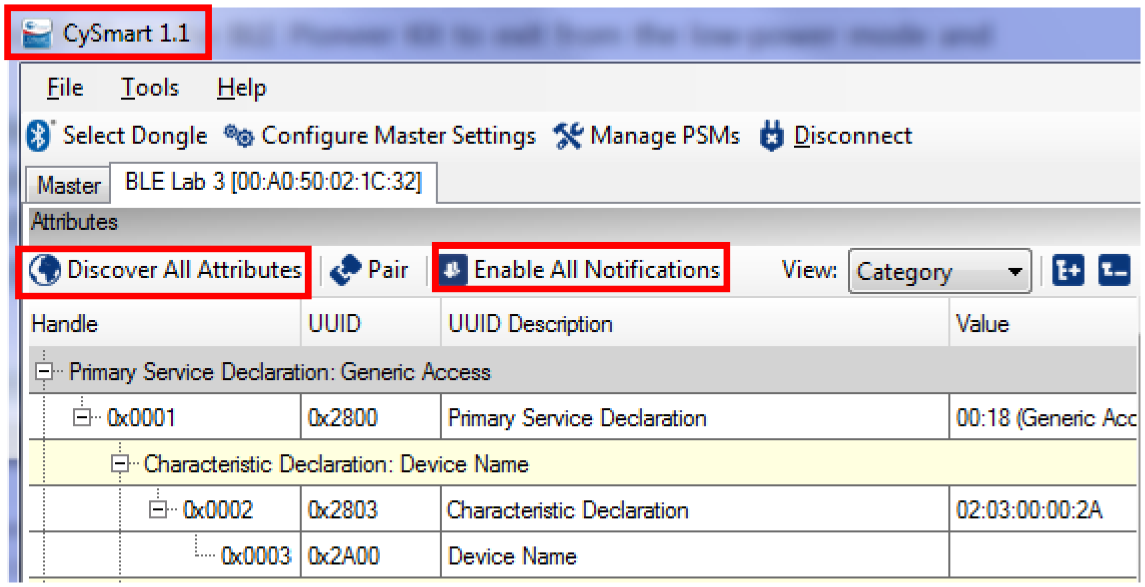


#	Device	Bluetooth Address	Address Type	RSSI	Advertisement Type	Cc
1	BLE Lab 3	00:A0:50:02:1C:32	Public	-35 dBm	Connectable undirected	

Figure 14: CySmart Scanning Results

3. Testing: HR notification

Upon connection, I clicked Discover All Attributes to list all the services, and then clicked Enable All Notification to enable Heart Rate Measurement Characteristic notifications. See Figure 15.

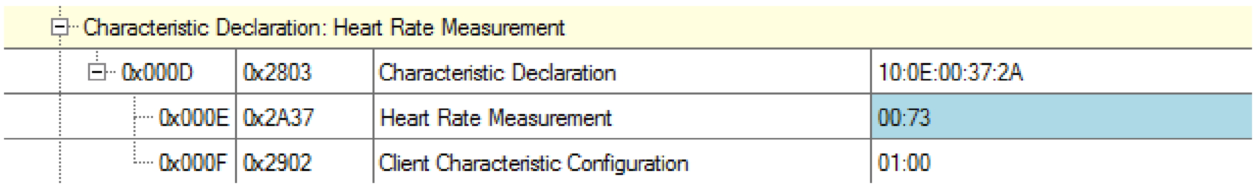


Handle	UUID	UUID Description	Value
Primary Service Declaration: Generic Access			
0x0001	0x2800	Primary Service Declaration	00:18 (Generic Acc
Characteristic Declaration: Device Name			
0x0002	0x2803	Characteristic Declaration	02:03:00:00:2A
0x0003	0x2A00	Device Name	

Figure 15: Heart Rate Characteristic Notifications

4. Testing: HR observation

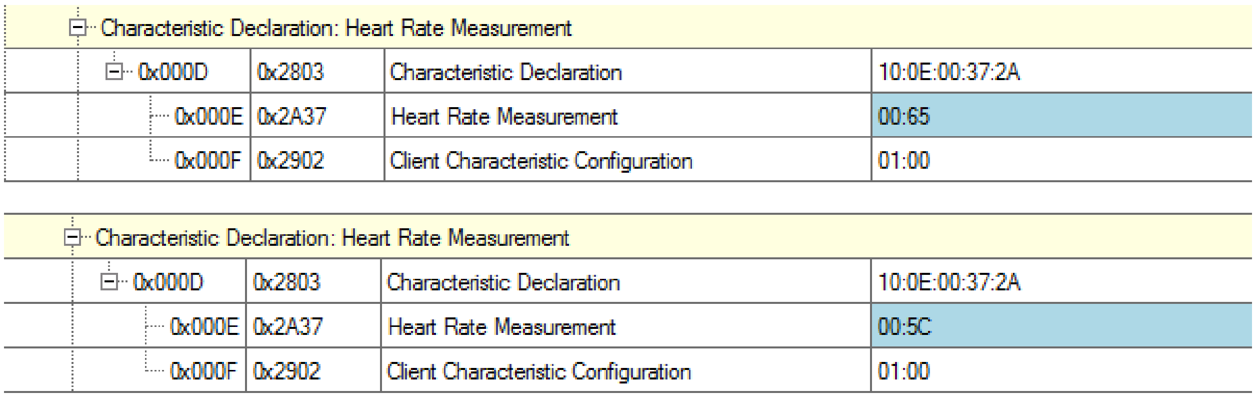
The value of Heart Rate is updated very second. See Figure 16.



Characteristic Declaration: Heart Rate Measurement			
0x000D	0x2803	Characteristic Declaration	10:0E:00:37:2A
0x000E	0x2A37	Heart Rate Measurement	00:73
0x000F	0x2902	Client Characteristic Configuration	01:00

Figure 16: Heart Rate Data in CySmart

When I pressed the SW2 switch on the kit, I found it will change the heart rate values within a range of 60 – 115 bpm, as shown in Figure 17.



Characteristic Declaration: Heart Rate Measurement			
0x000D	0x2803	Characteristic Declaration	10:0E:00:37:2A
0x000E	0x2A37	Heart Rate Measurement	00:65
0x000F	0x2902	Client Characteristic Configuration	01:00

Characteristic Declaration: Heart Rate Measurement			
0x000D	0x2803	Characteristic Declaration	10:0E:00:37:2A
0x000E	0x2A37	Heart Rate Measurement	00:5C
0x000F	0x2902	Client Characteristic Configuration	01:00

Figure 17: The results of pressing SW2

5. Testing: RGB LED observation

When the device is disconnected, the RGB LED turns off. Then, if I press the SW2 switch, the Green LED turns on indicating the device starts advertising again. When the device is connected with BLE, the Purple LED turns on showing the communication with BLE component. See the following Figure 18.

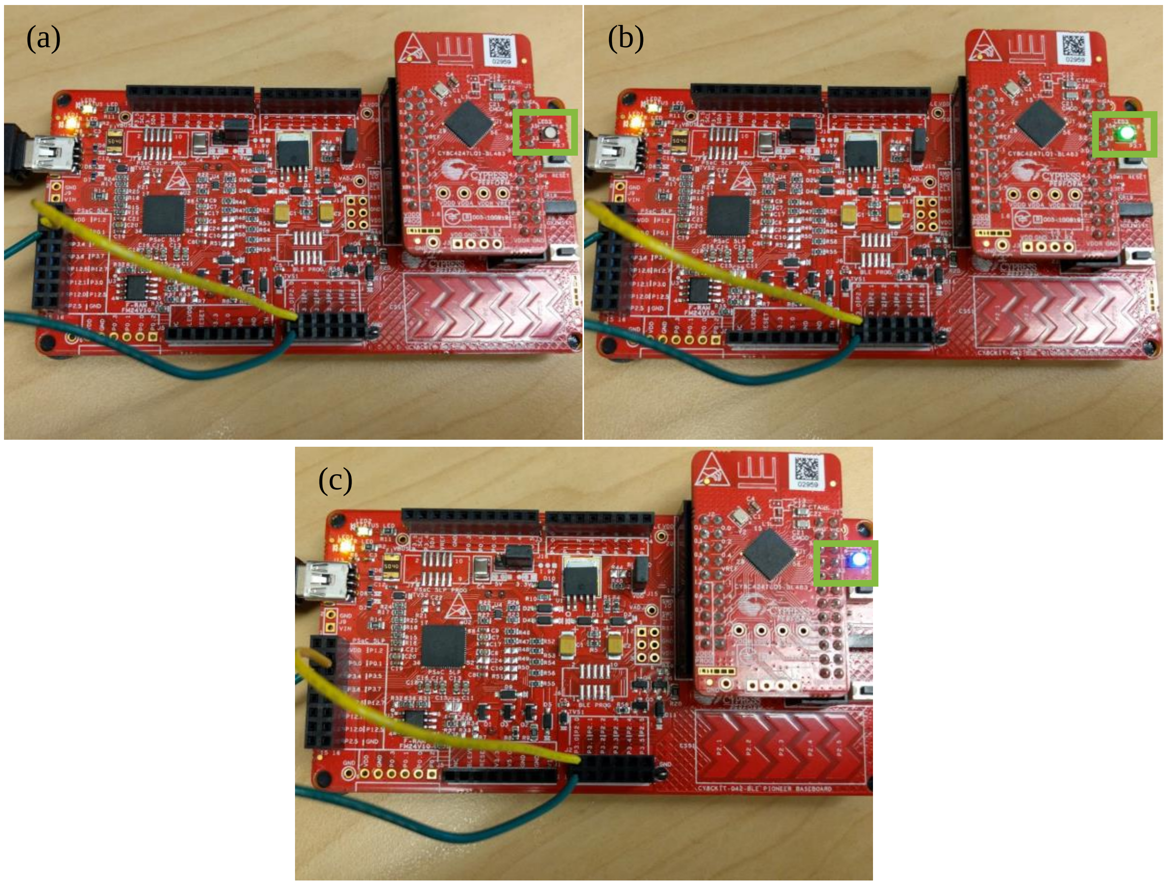
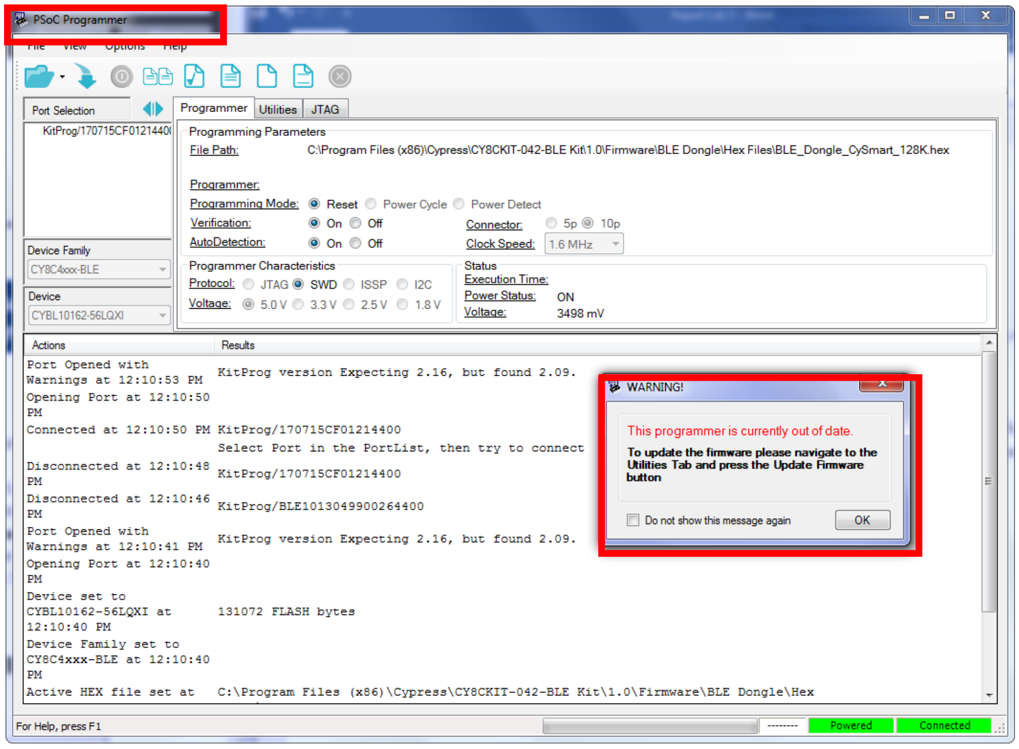


Figure 18: Different Status of LED such as (a) LED off in disconnection, (b) Green light in advertising, and (c) Purple light in communication

6. Restore PSoC 5

To restore the default firmware on PSoC 5, we need to use the PSoC Programmer. When it is opened, it detected the firmware installed on the PSoC 5 out of date and instructed me to upgrade the firmware. By following the instructions, I pressed the Upgrade Firmware button under the Utilities tab, as shown in Figure 19.



Programming Parameters
File Path: C:\Program Files (x86)\Cypress\CY8CKIT-042-BLE Kit\1.0\Firmware\BLE Dongle\Hex Files\BLE_Dongle_CySmart_128K.hex

Programmer:
Programming Mode: [✓ Reset] Power Cycle Power Detect
Verification: [✓ On] Off
AutoDetection: [✓ On] Off
Connector: 5p [✓ 10p]
Clock Speed: 1.6 MHz

Programmer Characteristics
Protocol: JTAG [✓ SWD] ISSP I2C
Voltage: [✓ 5.0 V] 3.3 V 2.5 V 1.8 V

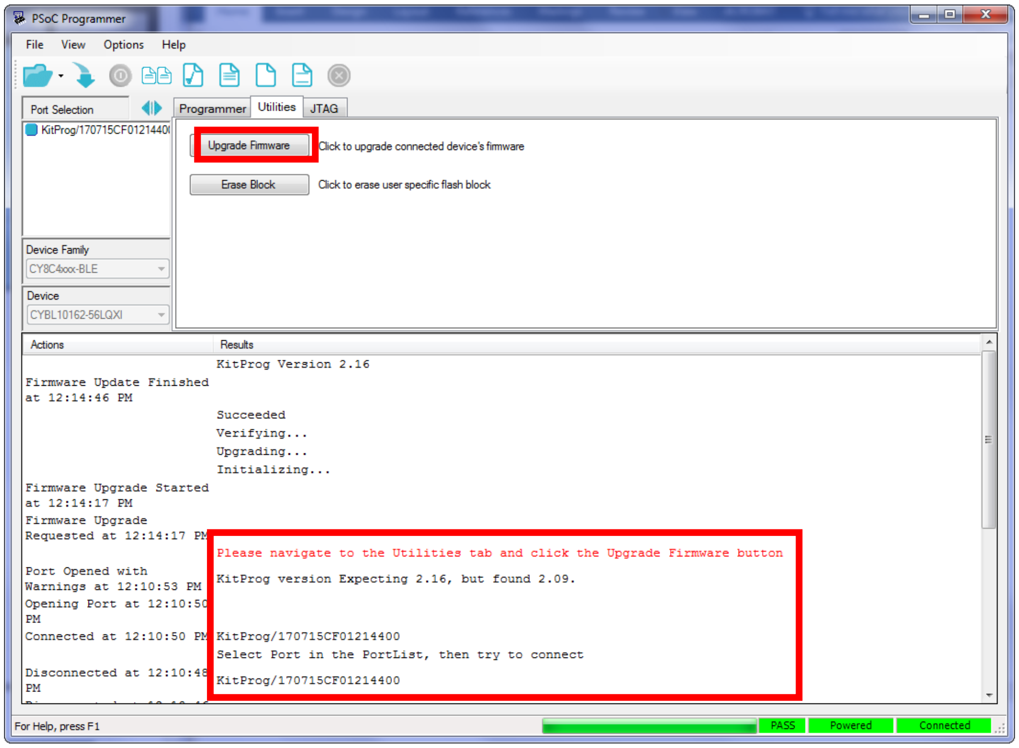
Status
Execution Time:
Power Status: ON
Voltage: 3498 mV

This programmer is currently out of date.

To update the firmware please navigate to the Utilities Tab and press the Update Firmware button

Do not show this message again

```
Actions                    Results
Port Opened with
Warnings at 12:10:53 PM    KitProg version Expecting 2.16, but found 2.09.
Opening Port at 12:10:50
PM
Connected at 12:10:50 PM KitProg/170715CF01214400
                           Select Port in the PortList, then try to connect
Disconnected at 12:10:48
PM                         KitProg/170715CF01214400
Disconnected at 12:10:46
PM                         KitProg/BLE1013049900264400
Port Opened with
Warnings at 12:10:41 PM    KitProg version Expecting 2.16, but found 2.09.
Opening Port at 12:10:40
PM
Device set to
CYBL10162-56LQXI at        131072 FLASH bytes
12:10:40 PM
Device Family set to
CY8C4xxx-BLE at 12:10:40
PM
Active HEX file set at     C:\Program Files (x86)\Cypress\CY8CKIT-042-BLE Kit\1.0\Firmware\BLE Dongle\Hex
```



Upgrade Firmware — Click to upgrade connected device's firmware

Erase Block — Click to erase user specific flash block

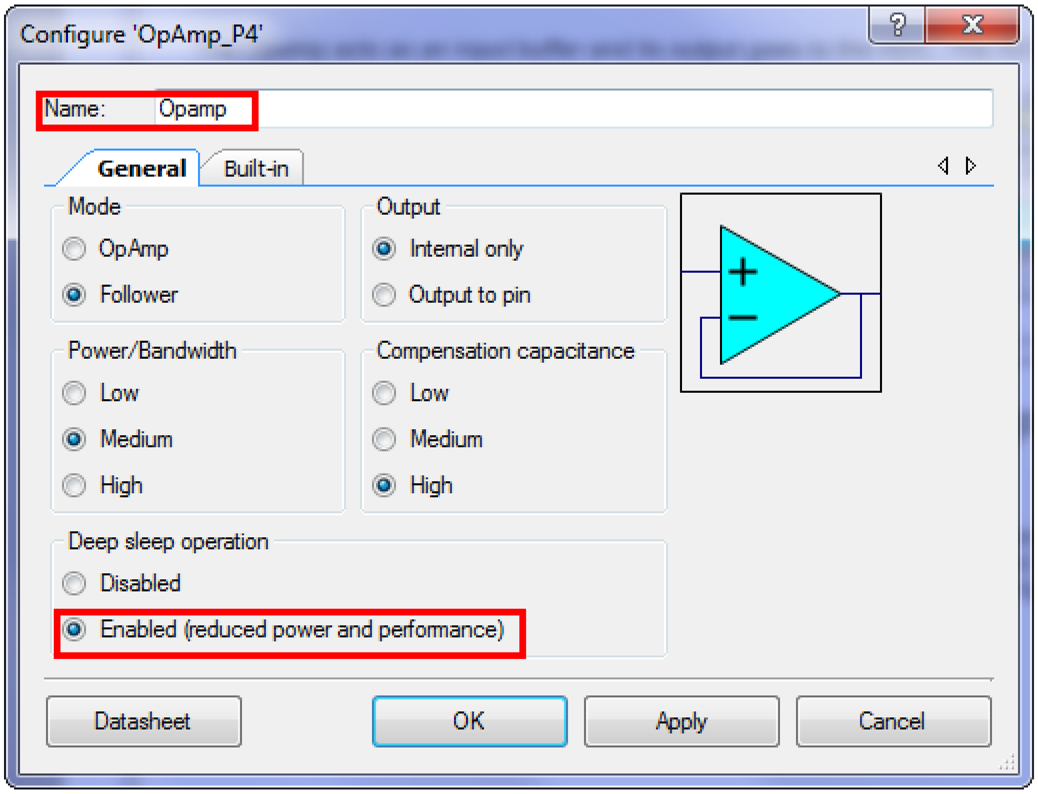
```
Actions                    Results
                           KitProg Version 2.16
Firmware Update Finished
at 12:14:46 PM
                           Succeeded
                           Verifying...
                           Upgrading...
                           Initializing...
Firmware Upgrade Started
at 12:14:17 PM
Firmware Upgrade
Requested at 12:14:17 PM
                           Please navigate to the Utilities tab and click the Upgrade Firmware button
Port Opened with
Warnings at 12:10:53 PM    KitProg version Expecting 2.16, but found 2.09.
Opening Port at 12:10:50
PM
Connected at 12:10:50 PM KitProg/170715CF01214400
                           Select Port in the PortList, then try to connect
Disconnected at 12:10:48
PM                         KitProg/170715CF01214400
```

Figure 19: Restore PSoC 5 in the PSoC Programmer.

ADDITIONAL EXERCISES

1. Opamp configuration: Deep-Sleep mode

Double click the Opamp component in the schematic and enable the Deep-Sleep operation.



Configure 'OpAmp_P4'

Name: Opamp

General | Built-in

Mode: OpAmp [✓ Follower]

Output: [✓ Internal only] Output to pin

Power/Bandwidth: Low [✓ Medium] High

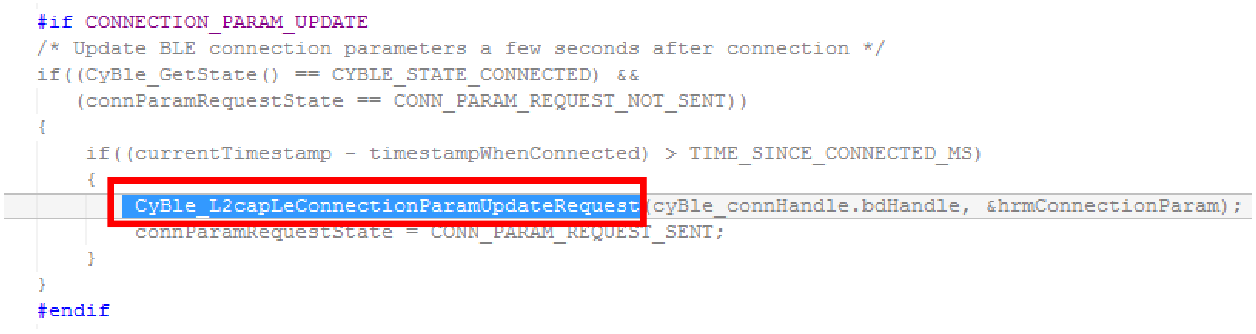
Compensation capacitance: Low Medium [✓ High]

Deep sleep operation: Disabled [✓ Enabled (reduced power and performance)]

Figure 20: PWM & Clock Configuration

2. Update the Connection Interval to 1 second

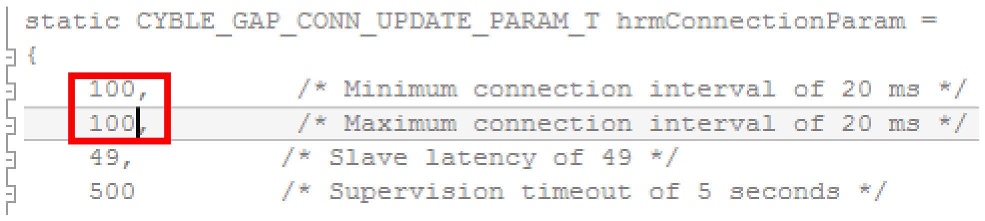
As shown in the main.c file, the API function CyBle_L2capLeConnectionParamUpdateRequest() is used to control the connection interval.



```
#if CONNECTION_PARAM_UPDATE
/* Update BLE connection parameters a few seconds after connection */
if((CyBle_GetState() == CYBLE_STATE_CONNECTED) &&
    (connParamRequestState == CONN_PARAM_REQUEST_NOT_SENT))
{
     if((currentTimestamp - timestampWhenConnected) > TIME_SINCE_CONNECTED_MS)
     {
          CyBle_L2capLeConnectionParamUpdateRequest(cyBle_connHandle.bdHandle, &hrmConnectionParam);
          connParamRequestState = CONN_PARAM_REQUEST_SENT;
     }
}
#endif
```

Figure 21: API function CyBle_L2capLeConnectionParamUpdateRequest()

I changed the parameter named hrmConnectionParam to set the connection interval to 1 second. The first two values should be set as 100. See Figure 22.



```
static CYBLE_GAP_CONN_UPDATE_PARAM_T hrmConnectionParam =
{
    100,          /* Minimum connection interval of 20 ms */
    100,          /* Maximum connection interval of 20 ms */
    49,          /* Slave latency of 49 */
    500          /* Supervision timeout of 5 seconds */
```

Figure 22: Change the parameter to 1 second

CONCLUSIONS

By completing Lab 3, I learned how to reload and upgrade the firmware on PSoC 5 to realize the function of a Heart Rate Signal. In addition, the kit can be developed into a Heart Rate Sensor System, and it is specifically designed to consume energy as low as possible. Thus, I investigated the low energy consumption strategy in PSoC 5 kit, where a continuous Active-Deep-Sleep power mode cycle is applied. It automatically goes back to Deep-Sleep after the completion of the corresponding Receive/Transmit, or enters the Hibernate mode in a long advertising state.

The additional exercises 1 shows how to configure the kit to a Deep-Sleep mode in order to consume less energy. The second additional exercise shows more configuration options like connection interval, which is useful to apply the system in a specific case. The last additional exercise shows another method to generate heart pulse signal. The process can be found in the old edition of Lab 3, where a TCPWM component inside the PRoC BLE module is used to simulate the heart rate signal.