PSoC 4 BLE Lab3: IoT Sensor-Based System Design

Group: Gloria Bauman and Kervins Nicolas

Link: https://github.com/gbauman01/Embedded\_Systems\_Lab\_3

**Due: October 14th, 2016**

**Description**

This lab teaches you to create a heart rate sensor device by measuring an analog signal input on the PSoC 4 BLE device and reporting the measured heart rate value to a BLE enabled device such as an iPhone.

**Objectives**

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

**Process: Configure Schematic**

1. Under the “TopDesign.cysch” from the “Workspace Explorer” drag and place a “Bluetooth Low Energy” Component from the “Component Catalog” under the communications category. This is in the “Bluetooth Low Energy tab” located at the bottom of the window.
2. Configure the component to the appropriate parameters and settings.
   1. Under the “General Tab”
      1. Profile: “Heart Rate”
      2. Profile Role: “ Heart Rate Server (GATT Server)”
   2. Under the “Profiles Tab”
      1. “Device Information” Section
         1. “Manufacturer Name String” Sub-Section
            1. Type in “Cypress Semiconductor” in the “Manufacture Name” value
   3. Under the “GAP Settings” tab
      1. “General” Section
         1. Check off “Silicon generated ‘Company assigned’ part of the address (00A050-XXXXXX)
         2. Device Name: “BLE Lab 3”
         3. Appearance: “Generic Heart Rate Sensor”
         4. MTU size (bytes): “23”
         5. TX power Level (dBM): “0”
      2. “Peripheral” Section
         1. “Advertisement Settings” Sub-section
            1. Discovery Mode: “General”
            2. Advertising Type: “Connectable undirected advertising”
            3. Filter policy: “Scan request Any | Connect request Any”
            4. Advertising channel map: “Any”
            5. Uncheck “Slow Advertising Interval”
         2. “Advertisement Packet” sub-section
            1. Check “Local name”

Local Name: Complete

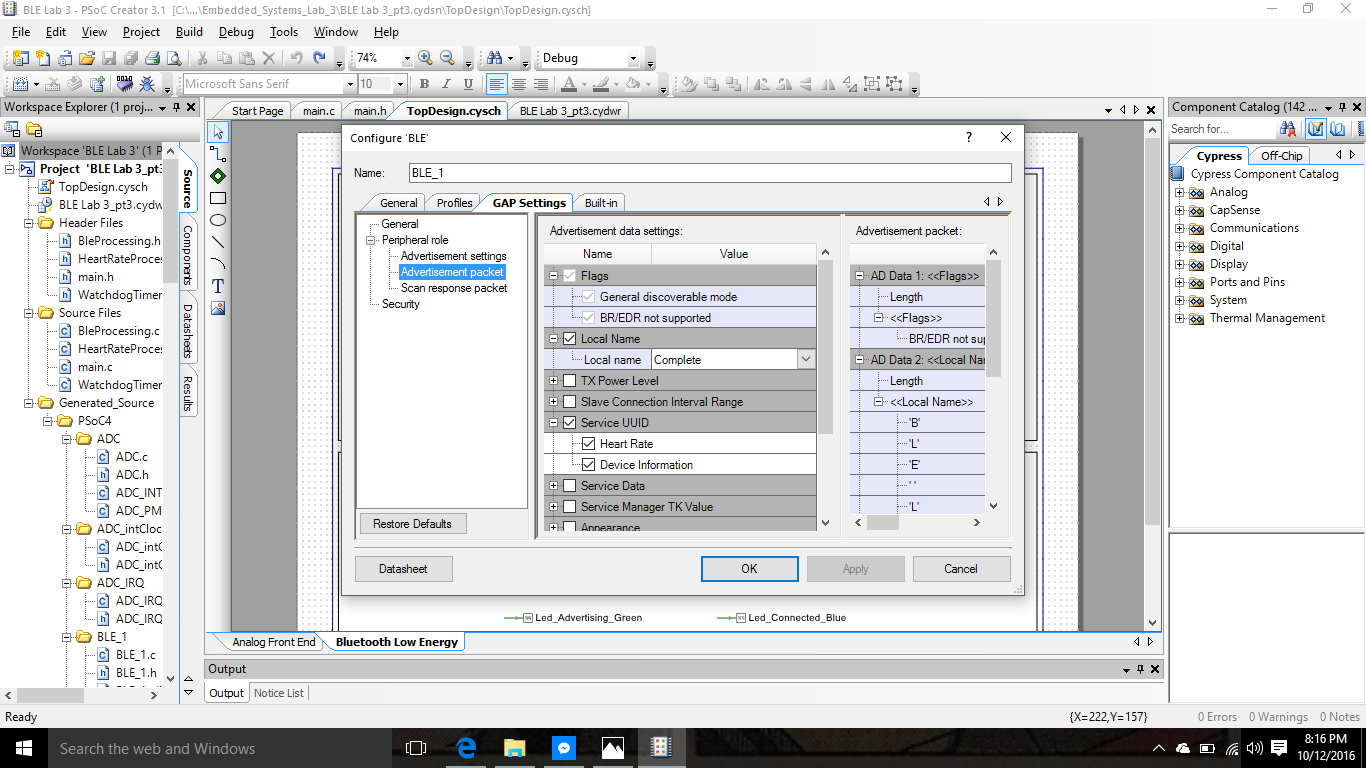
* + - * 1. Check “Heart Rate”

Check “Device Information”

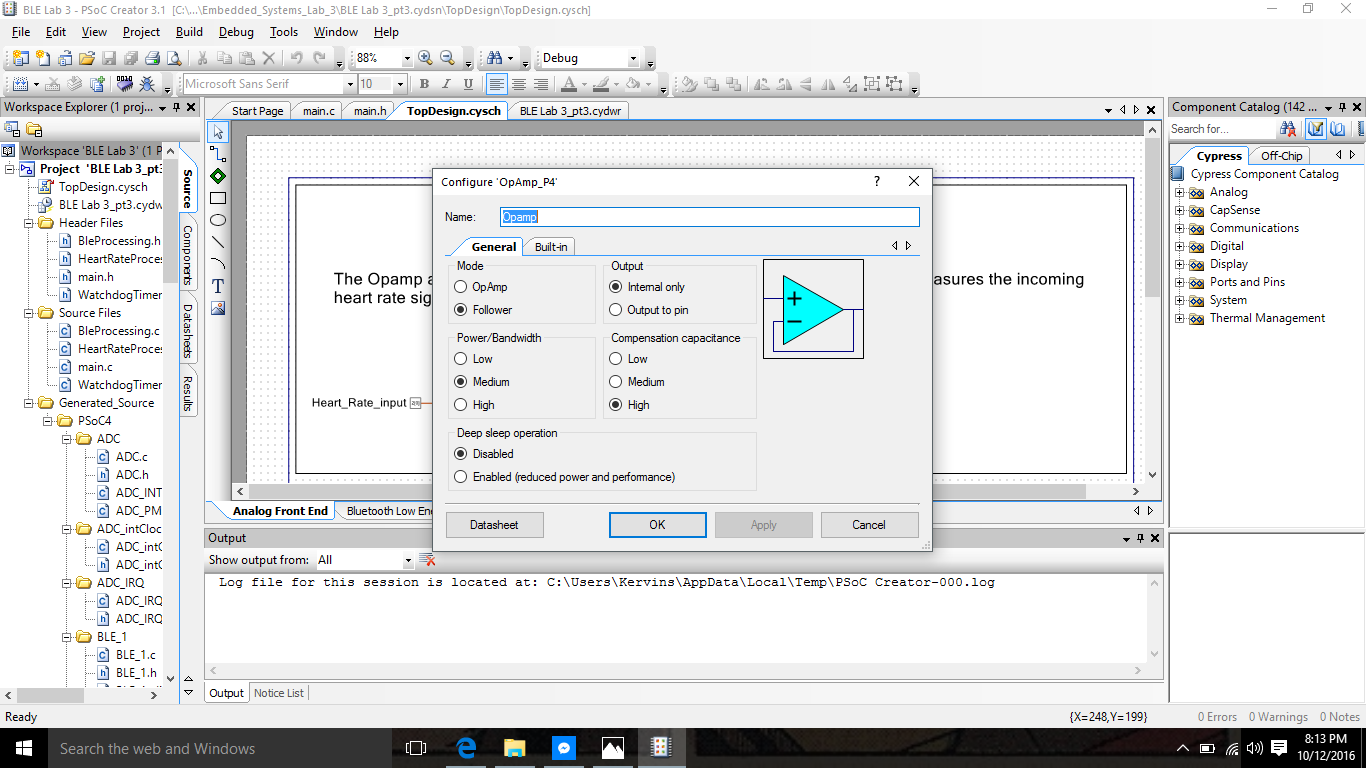
* + - 1. “Scan Response Packet” sub-section
         1. Check “Local Name”

Local Name: Complete

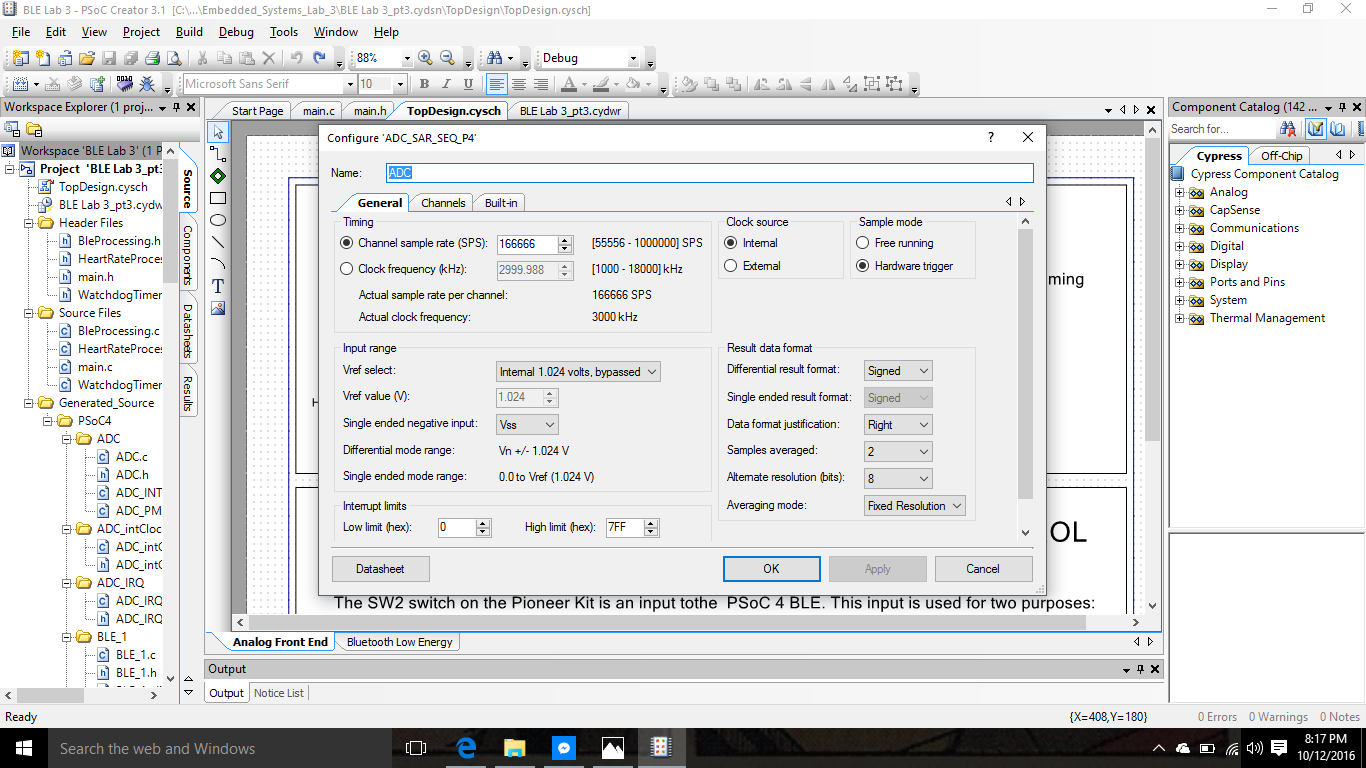
* + 1. “Security” Section
       1. Security Mode: “Mode 1”
       2. Security Level: “No Security(No authentication, No encryption)”
       3. I/O Capabilities: “No Input No Output”
       4. Paring Method: “Just Works”
       5. Bonding Requirement: “No Bonding”
       6. Encryption Key Size (bytes): “16”



1. Under the “Analog Front End” sheet of the schematic, and find the “Opamp” component in the Component Catalog. Drag the component and drop it onto the schematic, and double-click to configure it.
   1. Name: “Opamp”
   2. “General” Tab
      1. Check “Follower” under Mode
      2. Check “Medium” under Power/Bandwidth
      3. Check “Disabled” under Deep Sleep Operation
      4. Check “Internal Only” under Output
      5. Check “High” under Compensation Capacitance

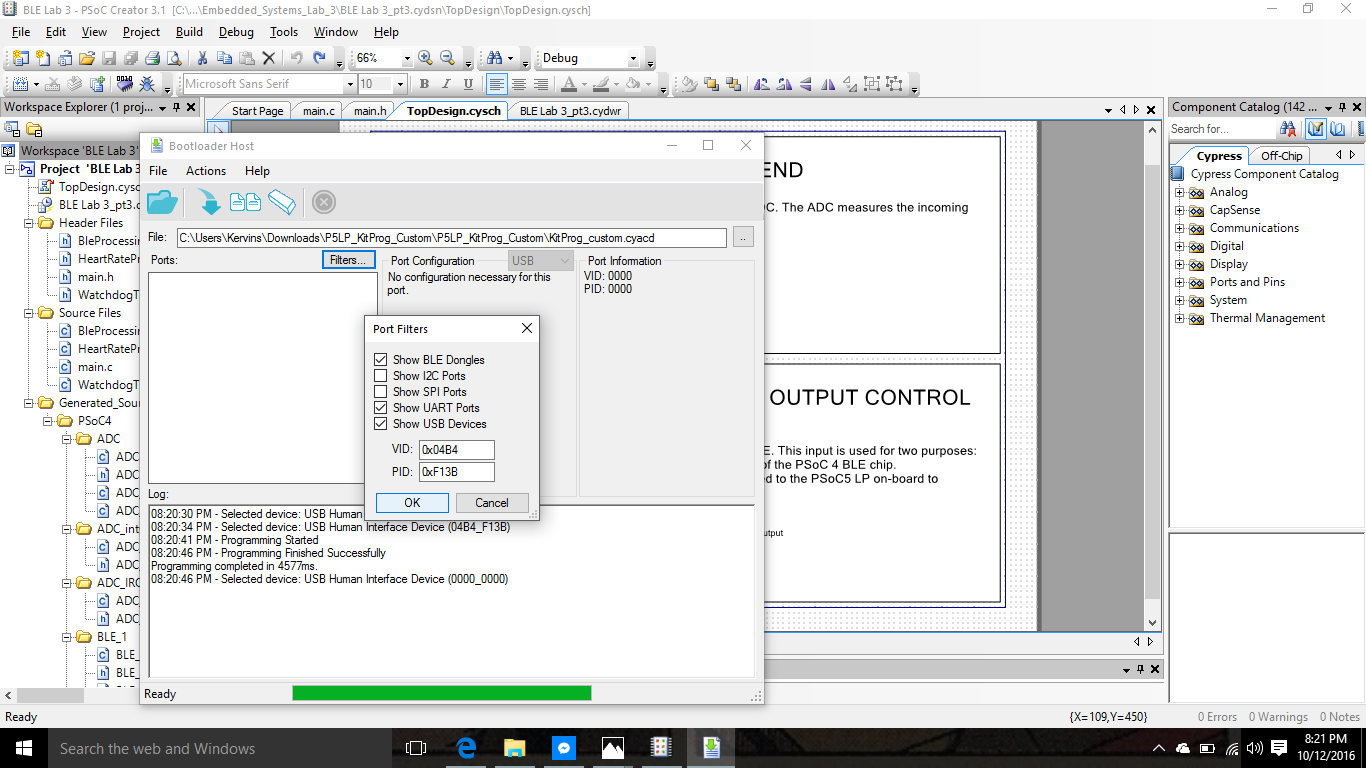


1. Find the “Sequencing SAR ADC” component in the component catalog. Drag it onto the “Analog Front End” sheet and configure the component.
   1. Name: “ADC”
   2. “General” Tab
      1. Timing
         1. Check “Channel sample rate (SPS)”: 166666
      2. Input Range
         1. Vref select: “Internal 1.024 volts, bypassed”
         2. Input buffer gain: “Disabled”
         3. Single ended negative input: “Yes”
      3. Clock Source
         1. Check “Internal”
      4. Sample Mode
         1. Check “Hardware trigger”
      5. Result Data Format
         1. Differential Result Format: “Signed”
   3. “Channels” tab
      1. Sequenced Channels: “1”
      2. Channel 0
         1. Check “enable”
         2. Resolution: 12
         3. Mode: Single
         4. AVG: unchecked
         5. ACq Time: “A clks”
         6. Conversion time: 6us
2. Add a “Low Logic ‘0’” component to the schematic editor, and connect its output to the “soc” input of the “ADC” component. Then connect the “Heart\_Rate\_Input” pin terminal to the ‘+’ input of the “ADC”



**Build and Program**

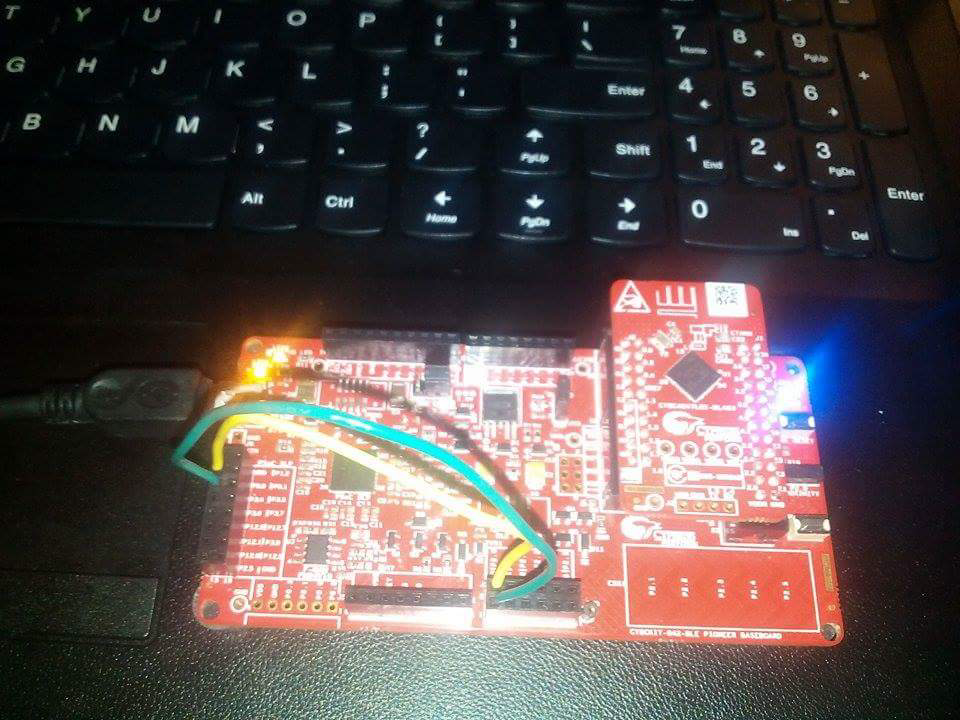
1. Bootloading PSoC 5: this will simulate a heart rate signal
   1. Remove the kit’s USB connector from PC
   2. While holding down “SW1 (reset)” plug in the kit’s usb connector to the PC, allowing the kit to enter the bootloader mode.
      1. Status LED (LED 2) starts blinking at a frequency of 1Hz.
   3. Launch “Bootloader Host Tools” through the “Tools” drop-down menu
      1. Click “Filters”
      2. Check “Show USB Devices”
      3. VID: “0x04B4”
      4. PID: “0x0F13B”
   4. After configuring the previous settings select the “USB Human Interface Device (04B4\_F13B)
   5. Open the bootloadable (\*.cyacd) file named “KitProg\_custom.cyacd”



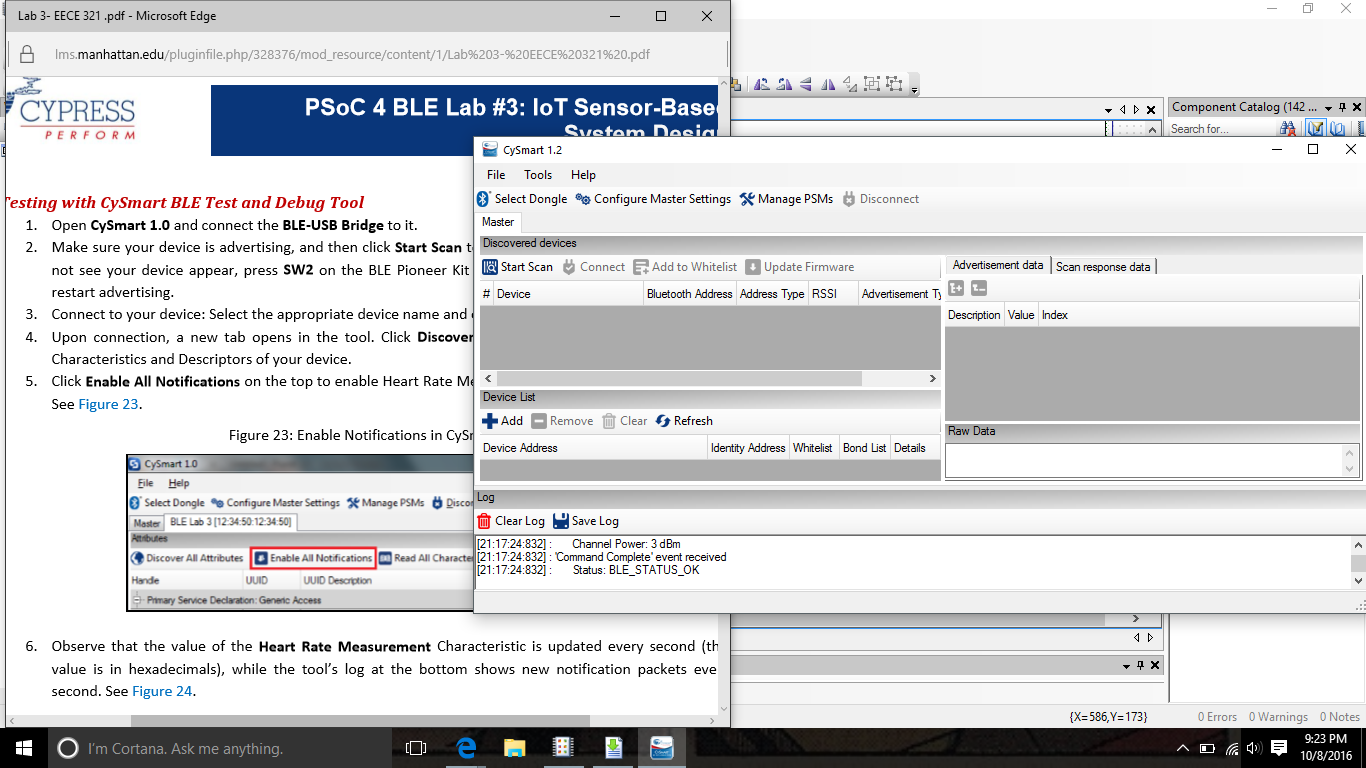
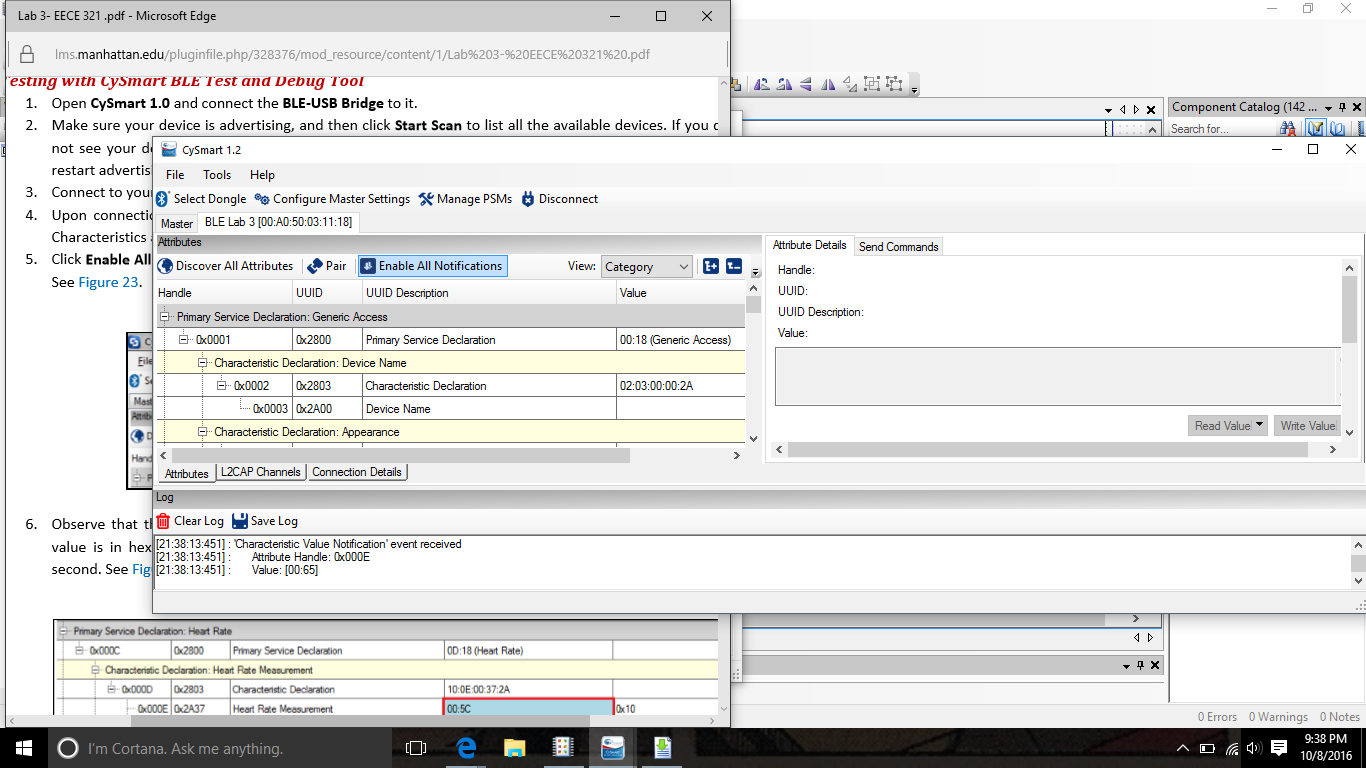
1. Program the PSoC 5 through the “Actions” drop-down menu

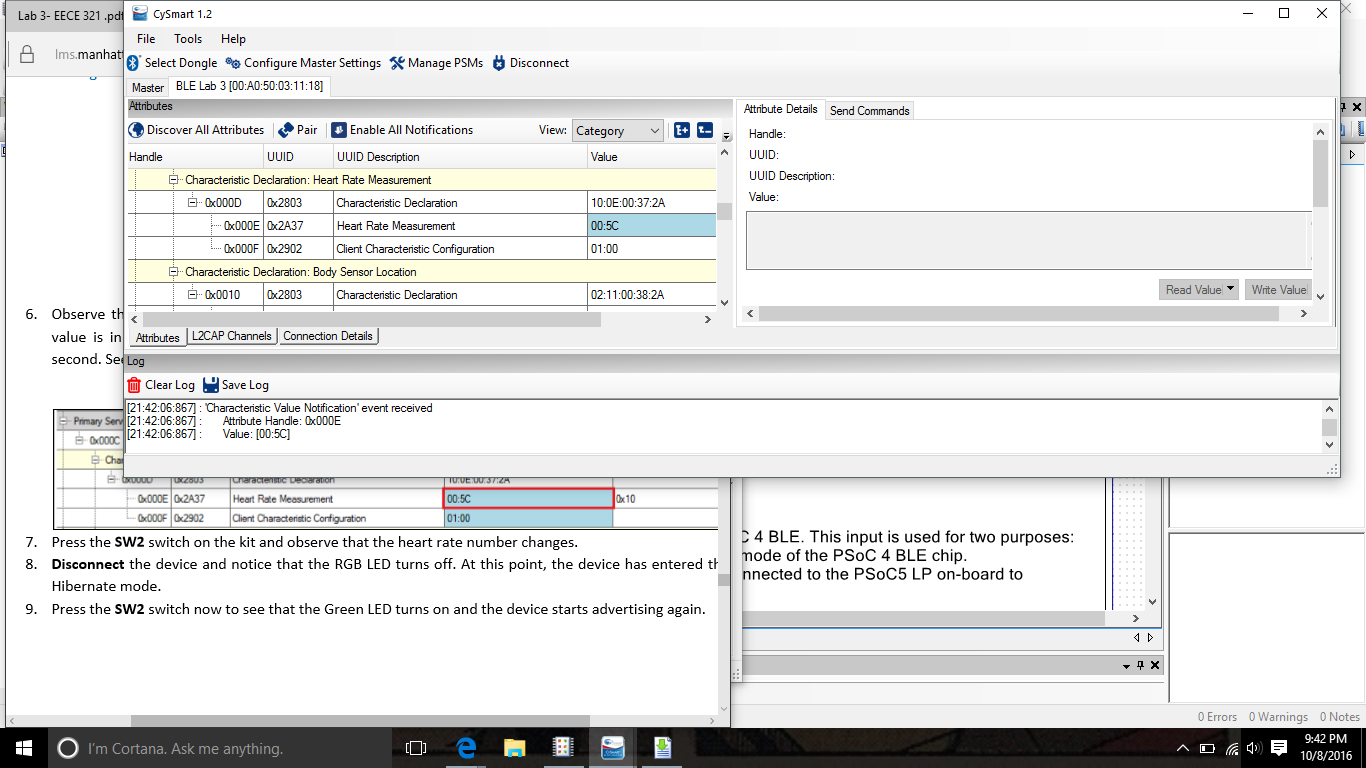
**Testing**

1. Bootloading the PSoC 5 simulated a heart rate signal on P0.0 with an expected value around 115. In order to see it connect that pin to “Pin 3” on “J8” to P2.0 of PSoC BLE “Pin 2” on “J2”
2. To change the heart rate value from a range of 60-115 bpm use “SW2” and connect PSoC 5 “Pin 5” on “J8” to PSoc BLE “pin 1” on “J2”
3. Check the status of the device by looking at the RGB LED

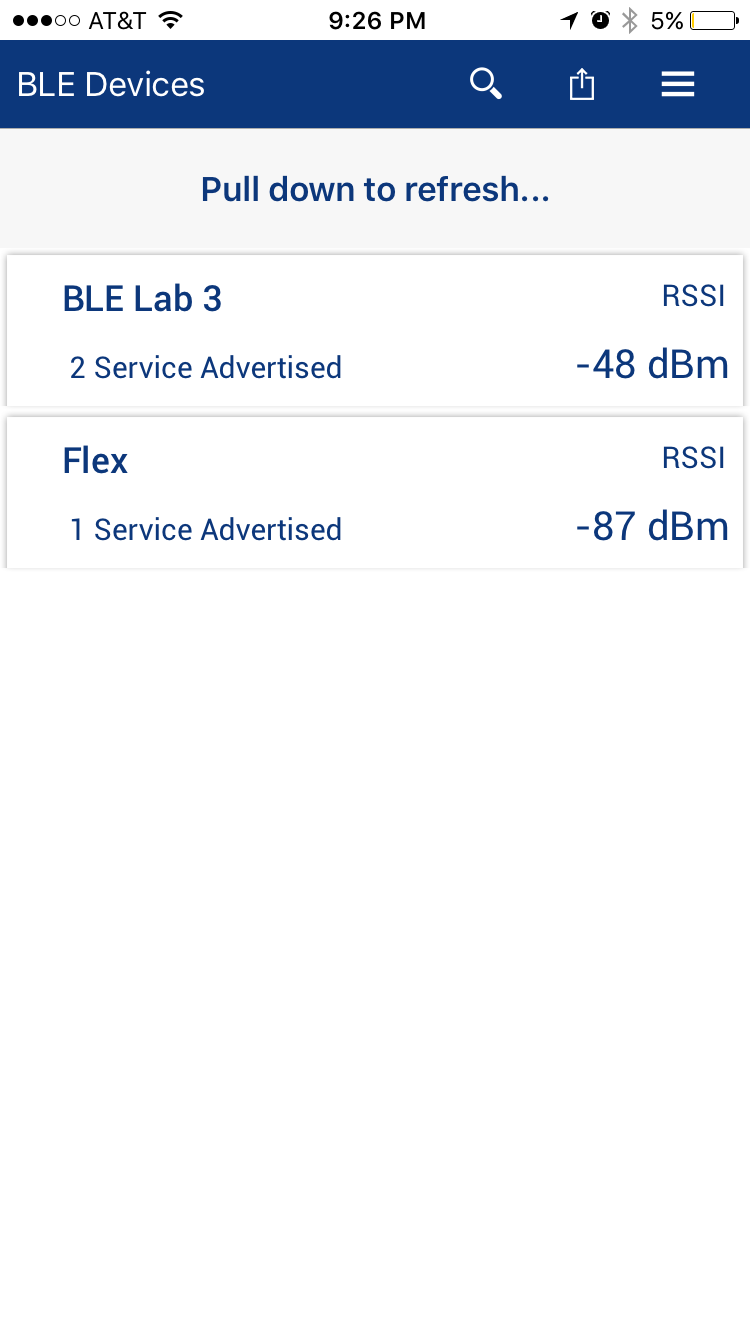


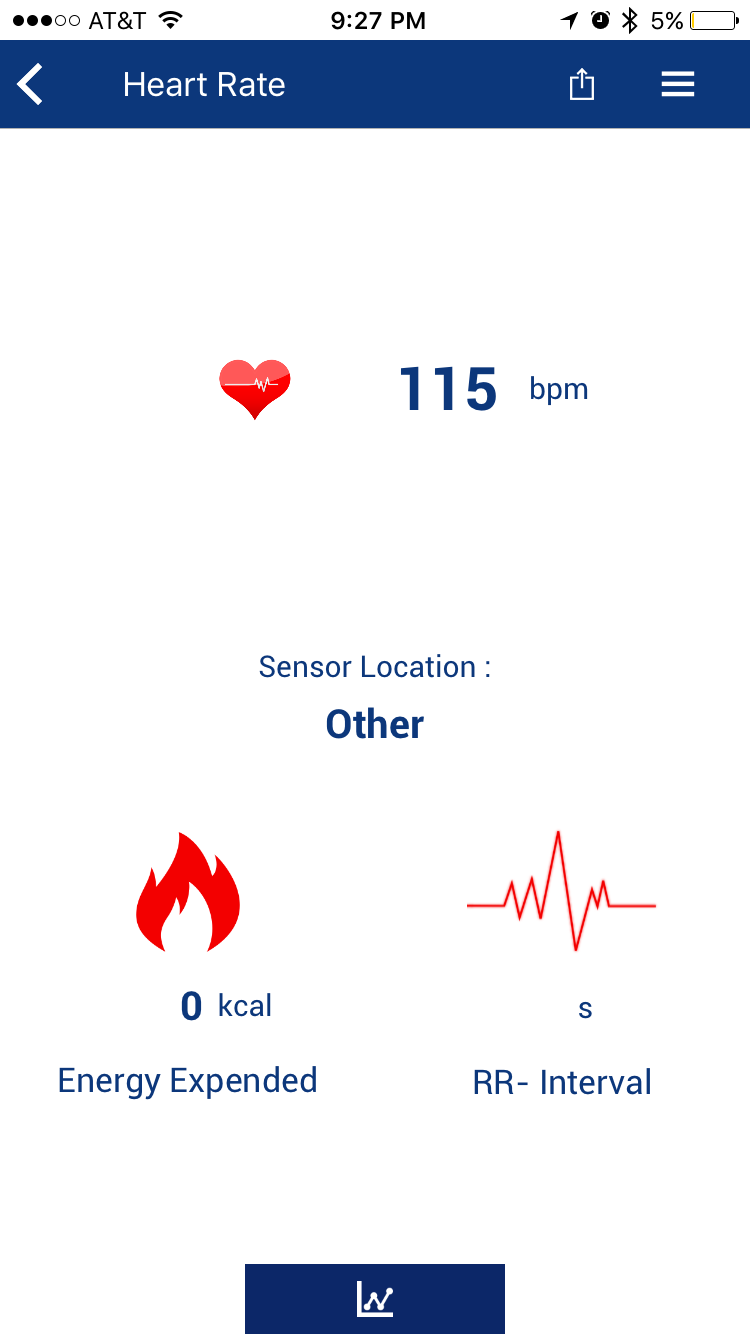
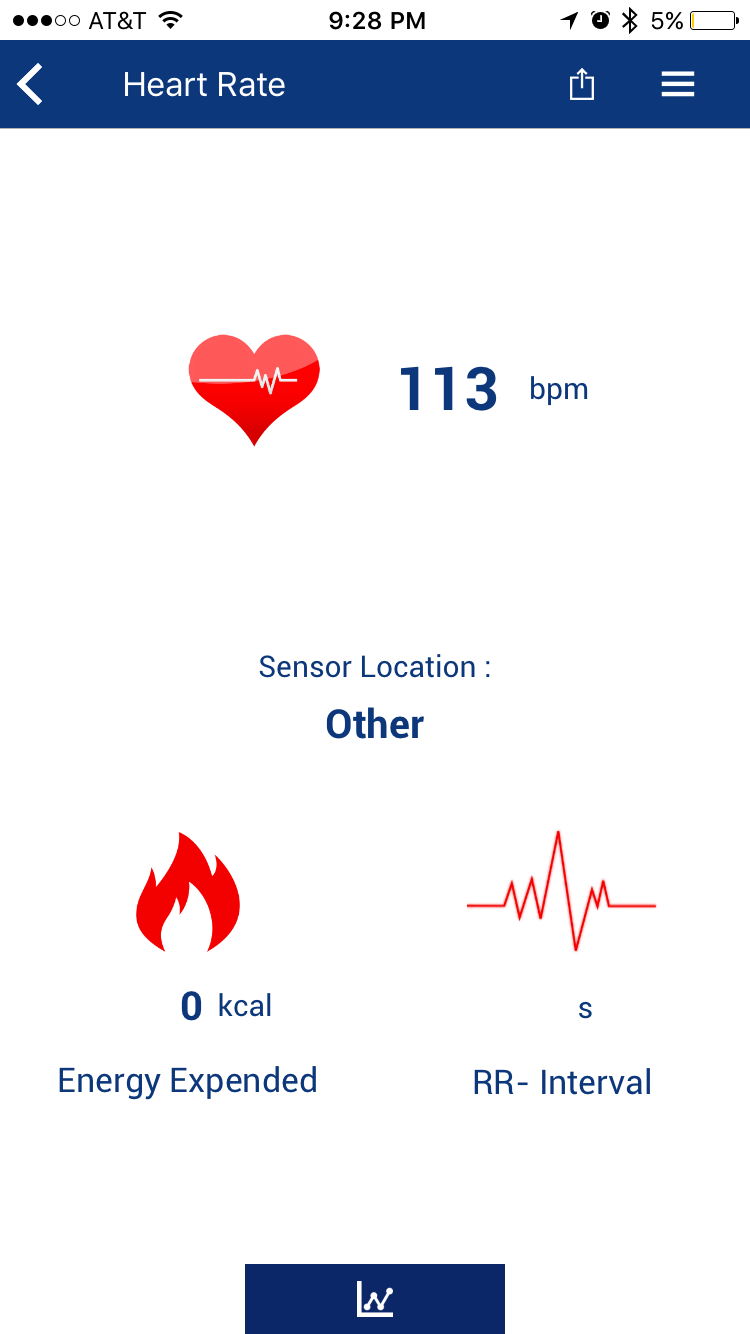
**Testing with CySmart BLE Test and Debug Tool**

1. Open CySmart 1.2 and connect it to the BLE-USB Bridge. Then “Starts Scan” (Press SW2 to restart advertising if it does not pop up) and connect your device.
2. Click “Discover All Attributes” and “Enable All Notifications”
   1. Observe “Heart Rate Measurement”
      1. 00:5C 
   2. Press “SW2” to change heart rate measurement



**Testing with CySmart BLE Test and Debug Tool**

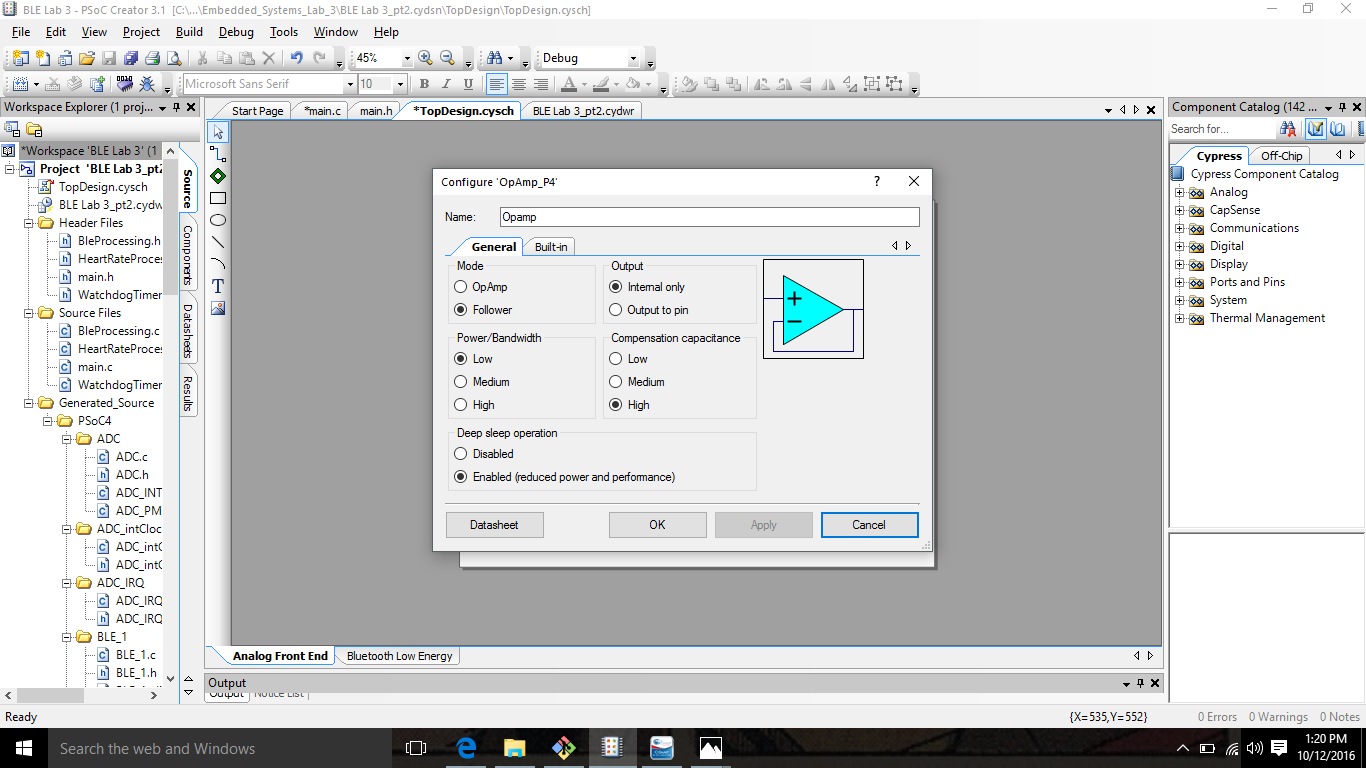
1. Open “CySmart Mobile App” on your mobile Bluetooth capable device
2. Find your device listed under the home screen as “BLE Lab 3” and tap on it.
3. Go to the “Profile” Screen and tap on the “Heart Rate” service.
4. Press “SW2” to change the heart rate. 



**Additional Exercises**

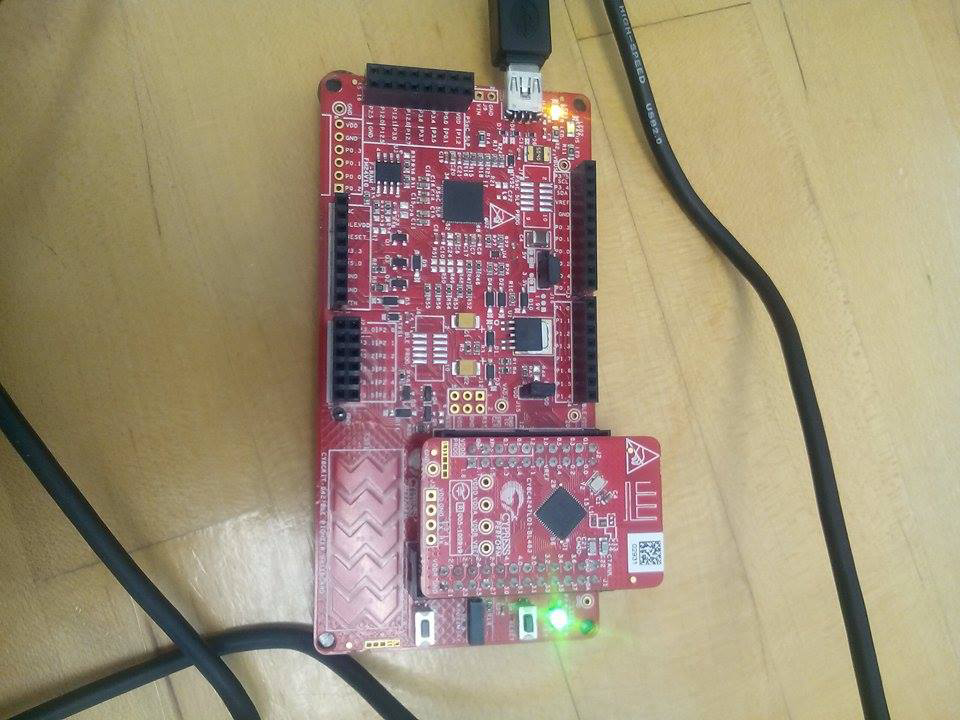
1. Configure the Opamp, used as a follower, to work in Deep-Sleep mode with lower power settings.

* Double-click the Opamp component
* Then change:
  + Mode to Follower
  + Power to Low
  + Enable the Deep Sleep Operation



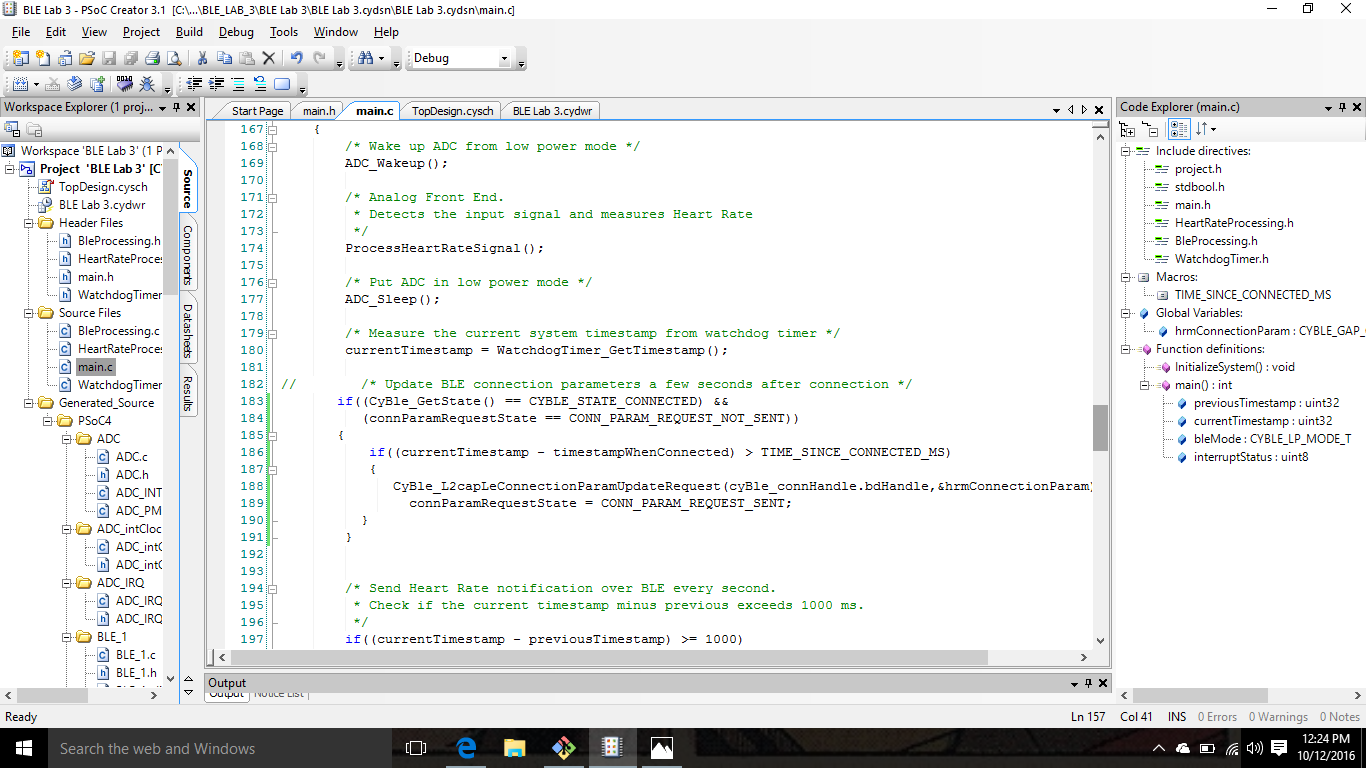
* Build the project

(No need to change the code)

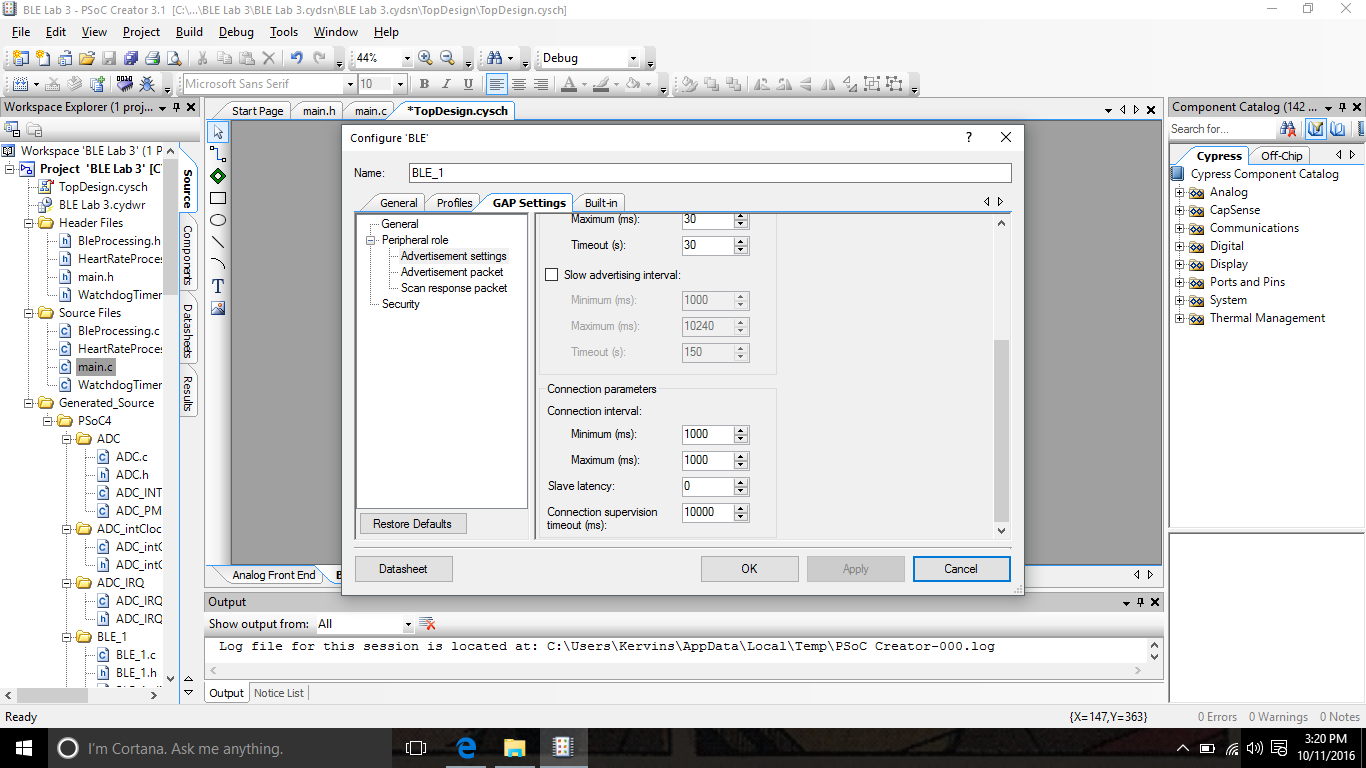


1. Update the Connection Interval to 1 second from the PSoC 4 BLE device.  
   Hint: Use the API function CyBle\_L2capLeConnectionParamUpdateRequest() to update the connection parameters. The API function is available in the completed firmware as commented code.

* Uncomment code in the main.c that update the BLE Connection

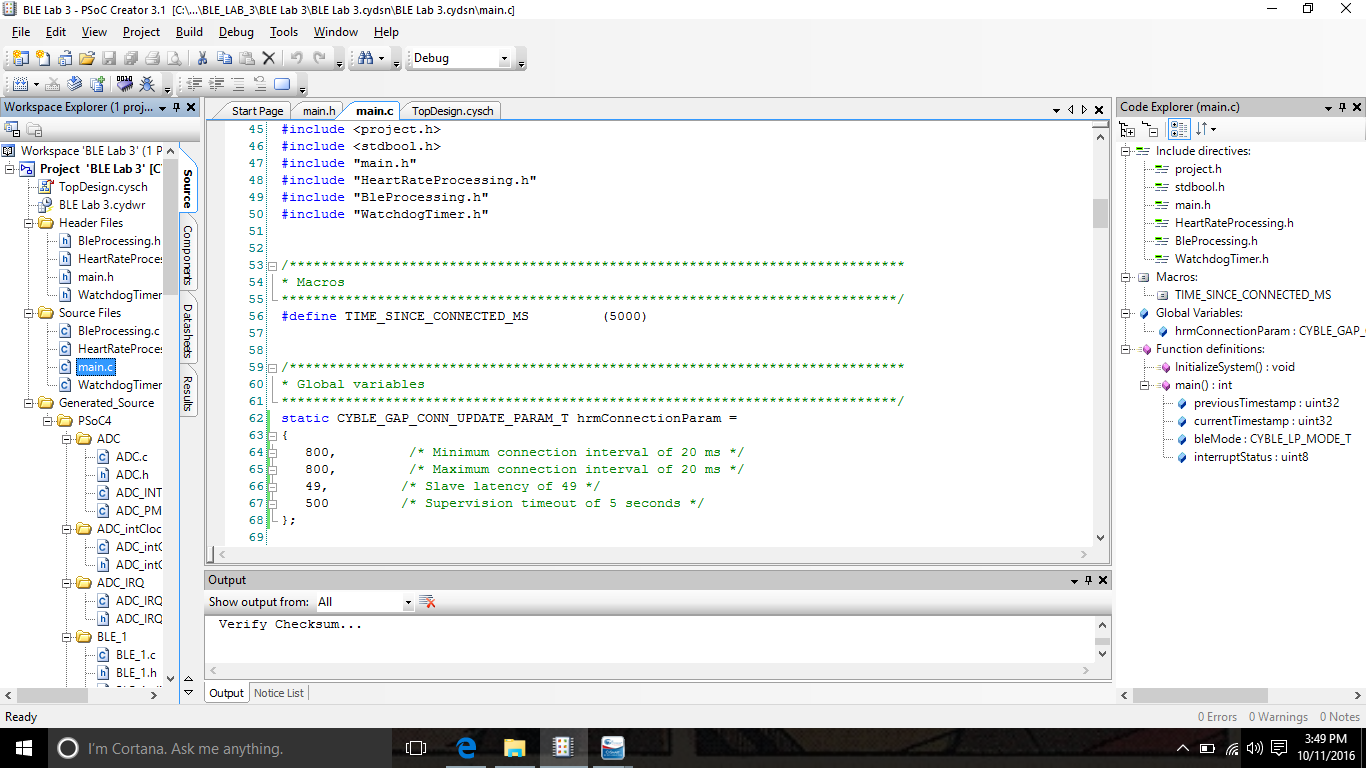


* Configure the BLE\_1 in the GAP Setting
* Go to the Connection Parameter
* Change the Min and Max values of the Connection Interval to 1000ms

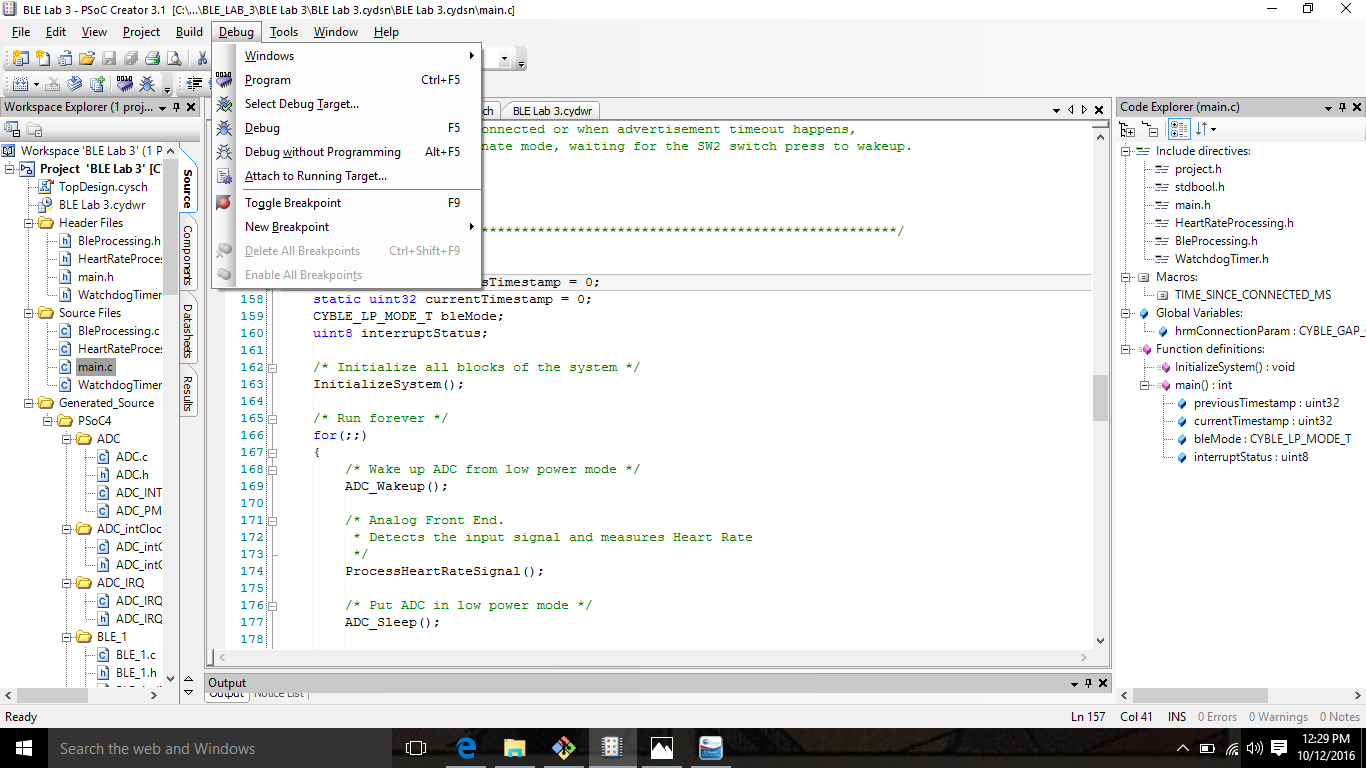


* Locate Global Variables in the main.c
* Uncomment the code for the connection parameter
* Must configure the Min and Max values of the Connection Interval of 20ms into a number that will allow the connection intervals equal to1000

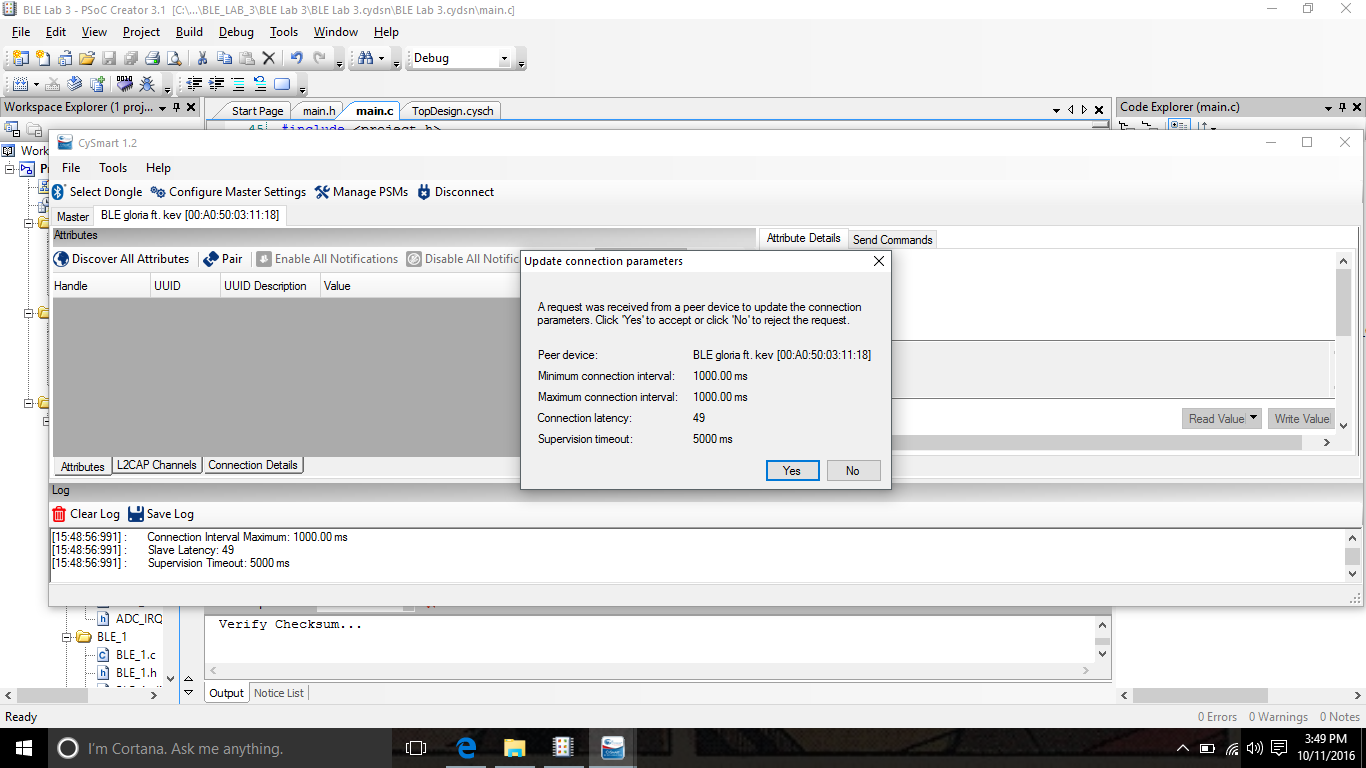
(Number = 800;)



* Debug the program

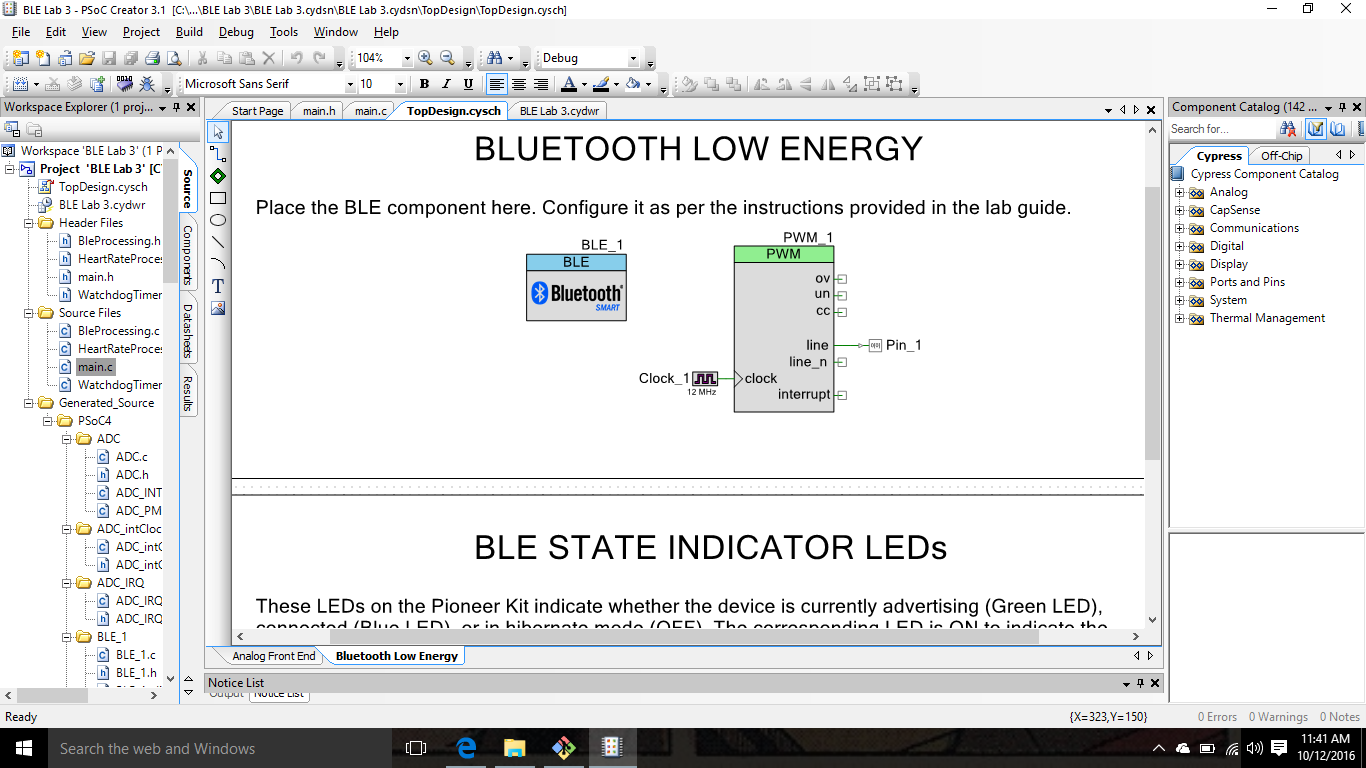


* Connect the BLE Dongle
* Open Cysmart 1.2
* Connect to your project file
* Word Box should pop up after giving information on device, stating the Connection Interval

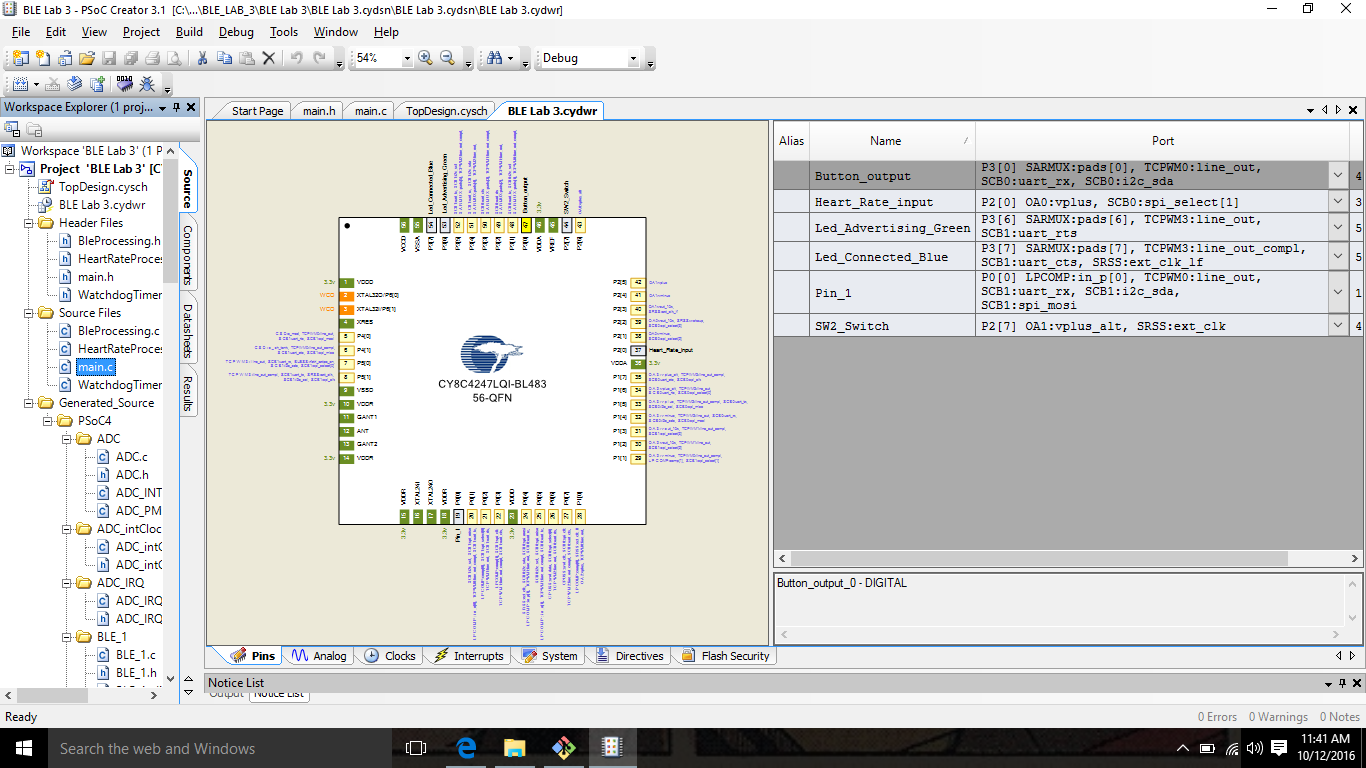


1. Generate the heart rate pulses using a PWM Component inside the PSoC 4 BLE.  
   Hints:  
    Don’t forget to start the PWM Component using the <component\_name>\_Start() API.  
    As the TCPWM block is not active during the Deep-Sleep mode, comment out the code for putting the device into the Deep-Sleep mode.

* Add a PWM component into the Bluetooth Low Energy
* Add a Clock component
* Connect to PWM clock
* Add Pin Output component
* Connect Pin Output to PWM line

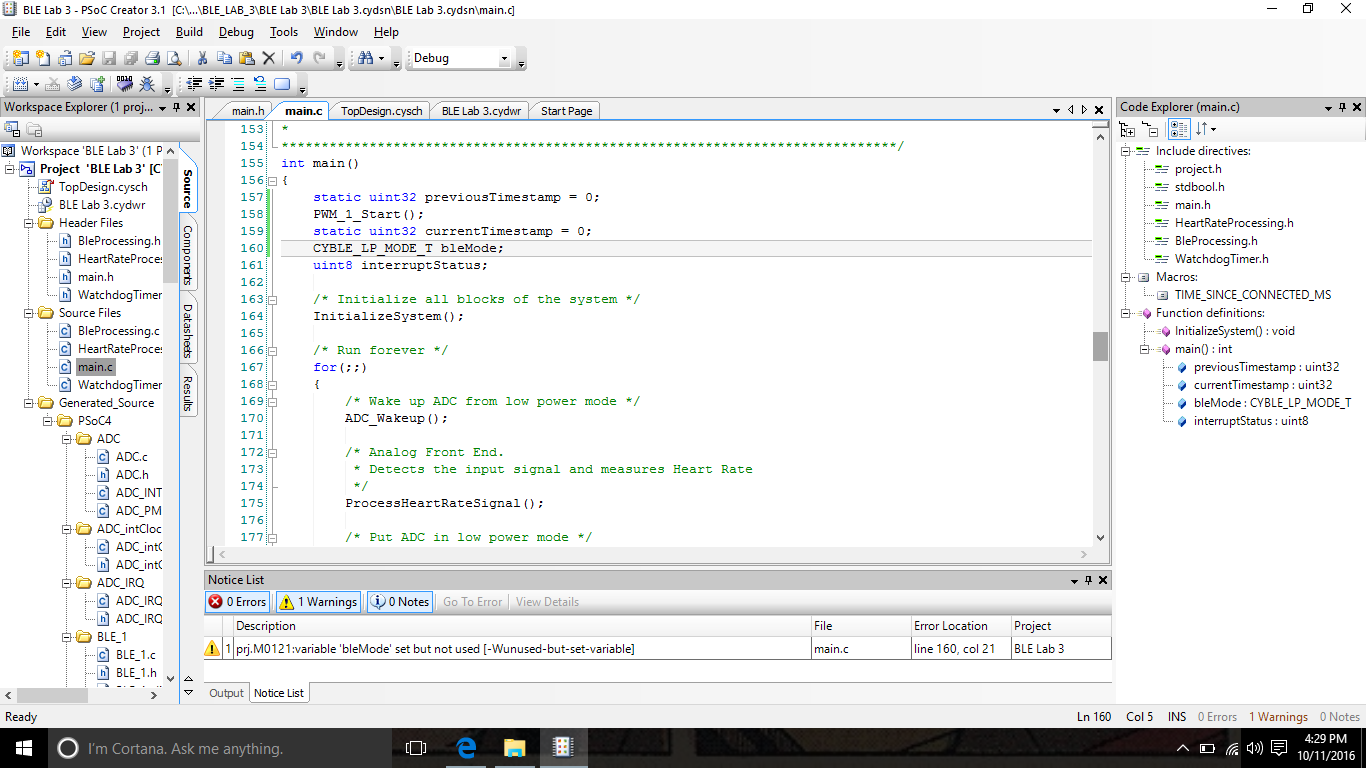


* Initialize Pin\_1 in .cydwr
* Change port to P0[0] for the Pin



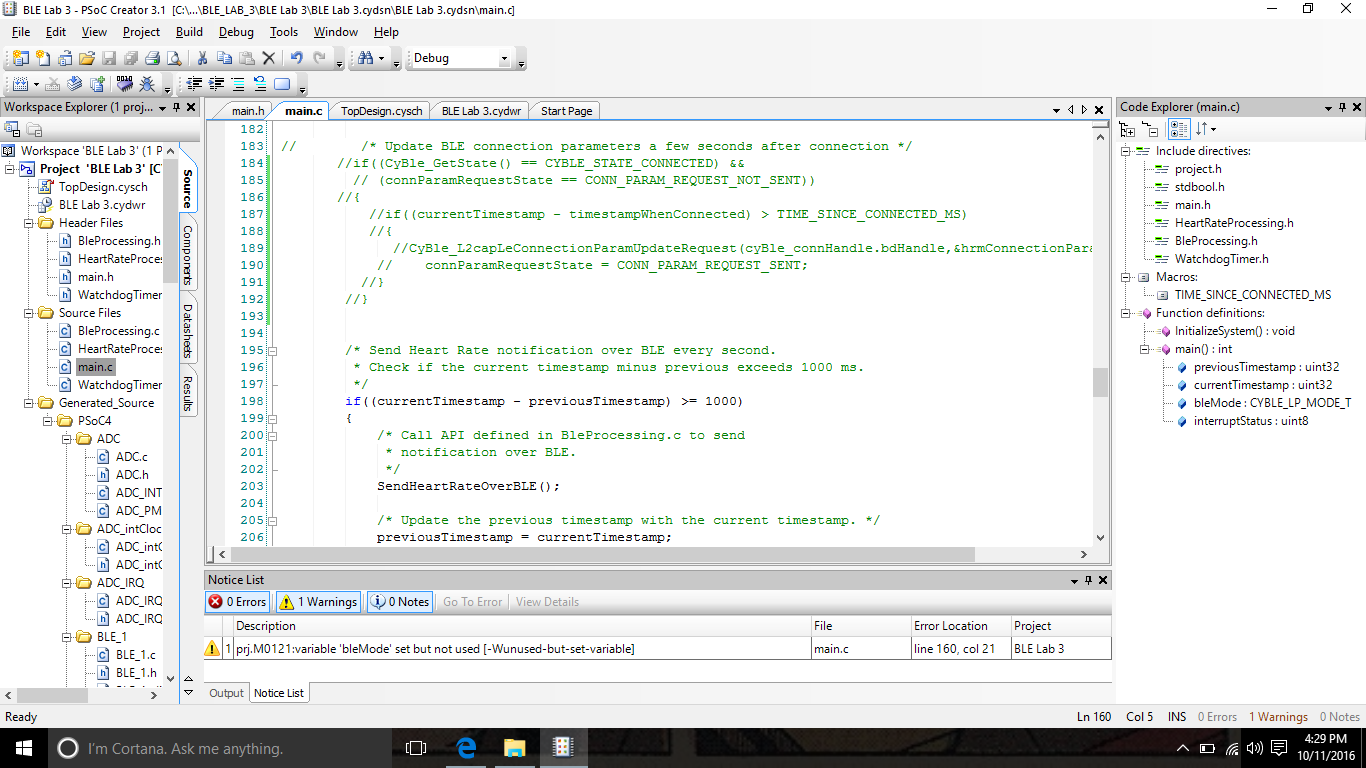
* Initialize the PWM in the code:

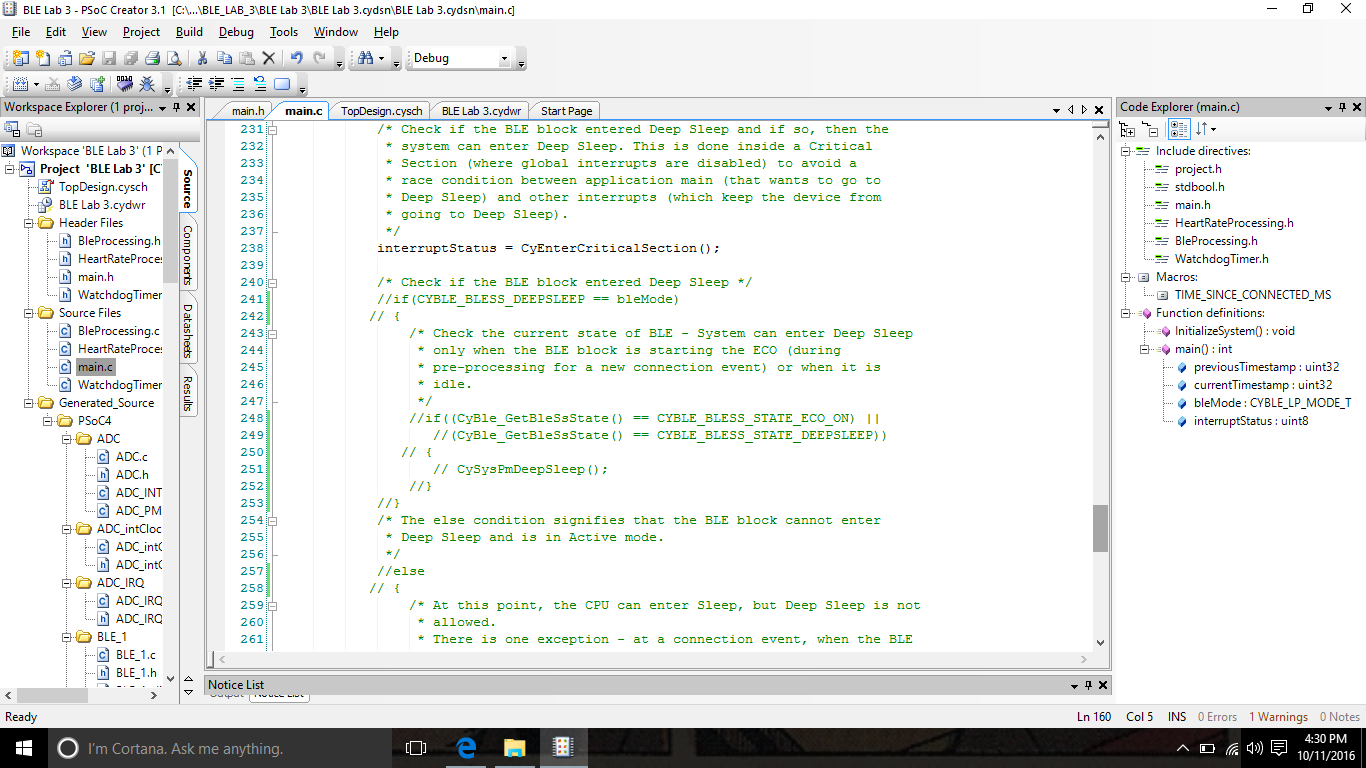
PWM\_1\_Start();

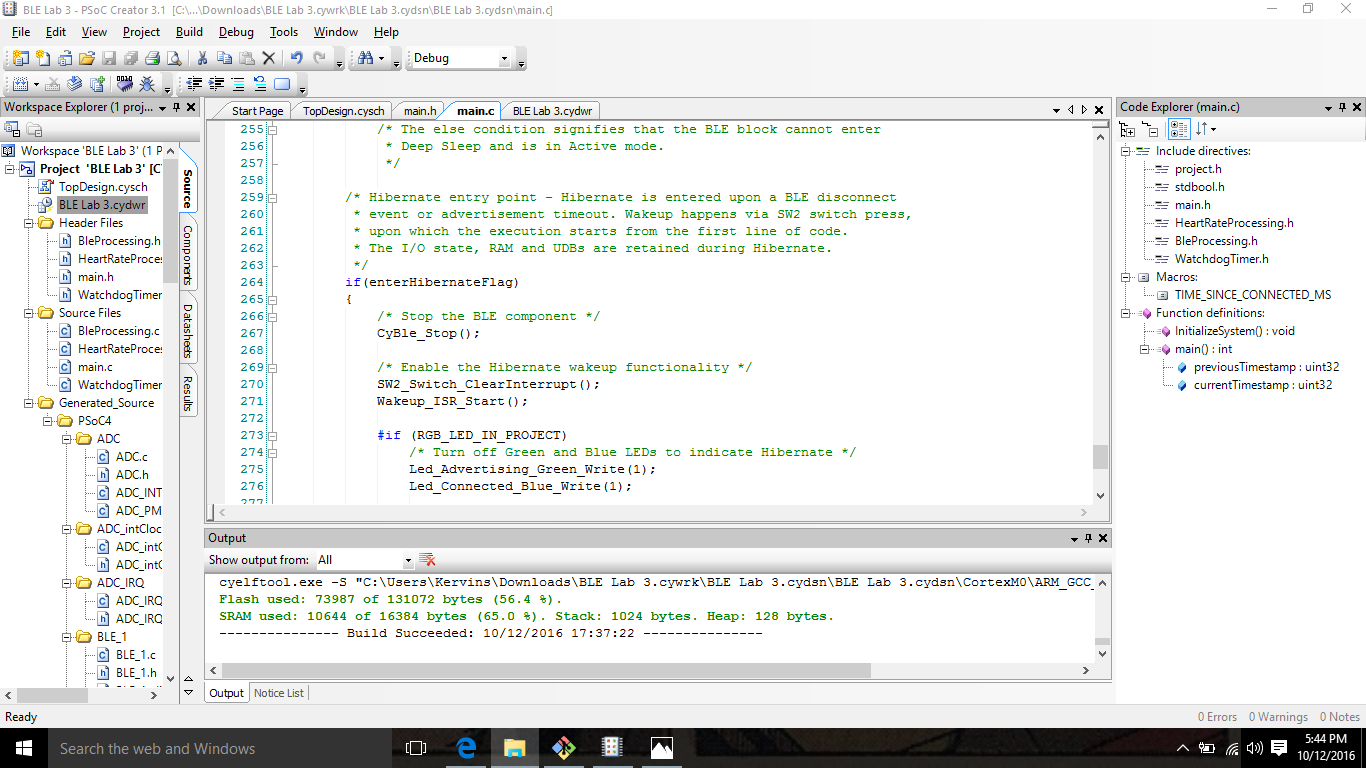


* Comment out all code regarding Deep-Sleep mode

(Don't want to put device in Deep-Sleep mode)







* Debug Program
* Build Project
* Open the Cysmart App on phone

Refresh until you find project

Go to Heart Rate screen

14672886_1605607972798781_1264169893_o.png