Lab 3

IoT Sensor-Based System Design

Joseph Lundgren and Louis Cooper

Manhattan College

9 October 2016

Embedded Systems Design

Introduction

The purpose of this lab is to configure the Pioneer Kit as a BLE GATT Server and heart rate sensor. Using a Digital Input Pin to simulate an analog signal. We will use an Android cell phone as the client which will receive the reported hear rate value.

Procedure

First, the lab asks to add a BLE component and configure it (figure 1). PSoC creator includes a custom Heart Rate profile which we used for this lab. We needed to configure the Device Information Service on the BLE component. This is important because this service is what tells the client (in this case our cell phone), what kind of BLE device that is communicating with (i.e. BLE Pioneer Kit).

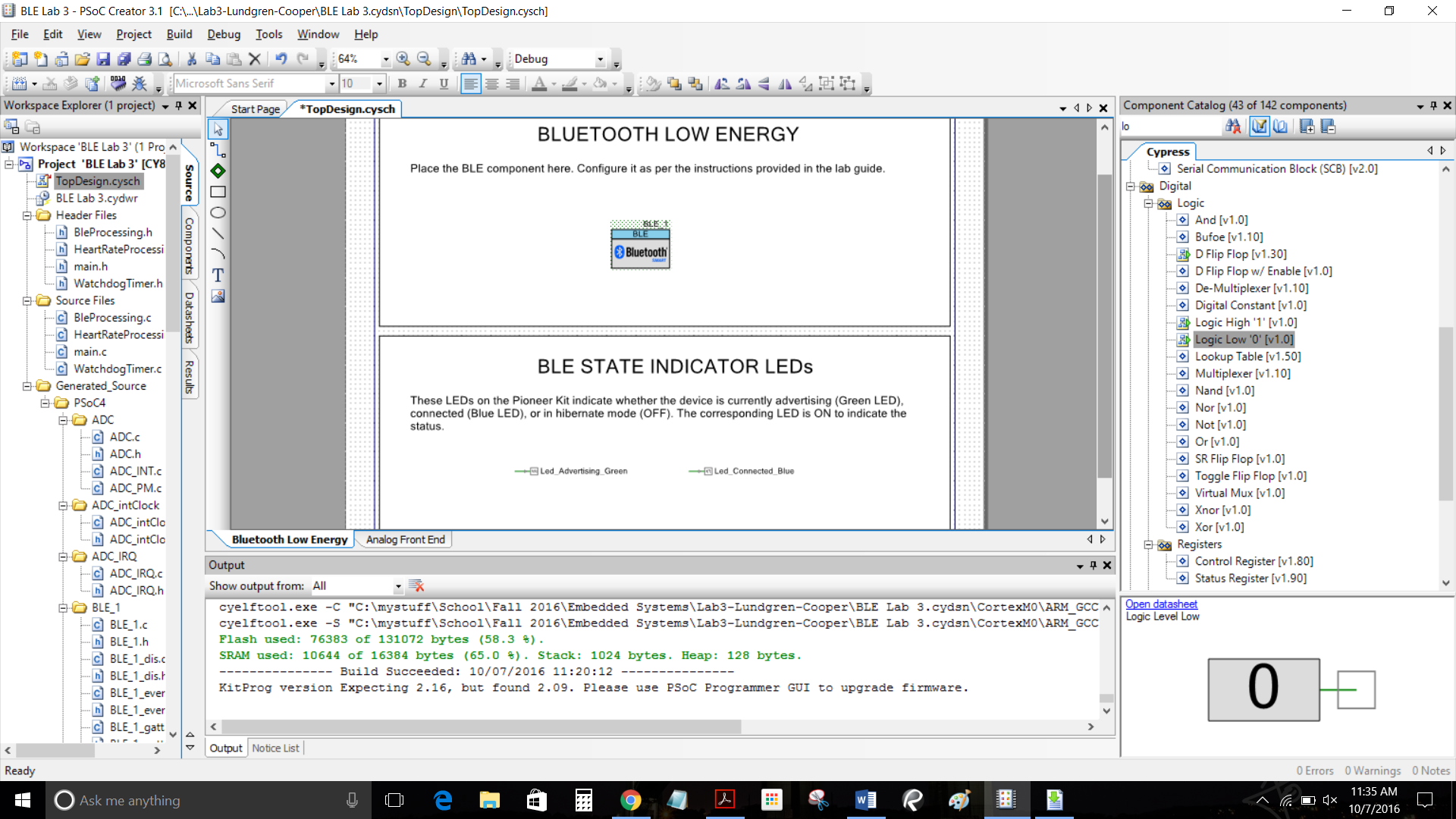


Figure 1

The final components needed for the operation of this lab are an Opamp, Sequencing SAR ADC, and the Logis Low ‘0’ (see figure 2). The Opamp is added to the workspace and configured. The Logic Low ‘0’ component is connected to the reset pin of the SAR ADC.

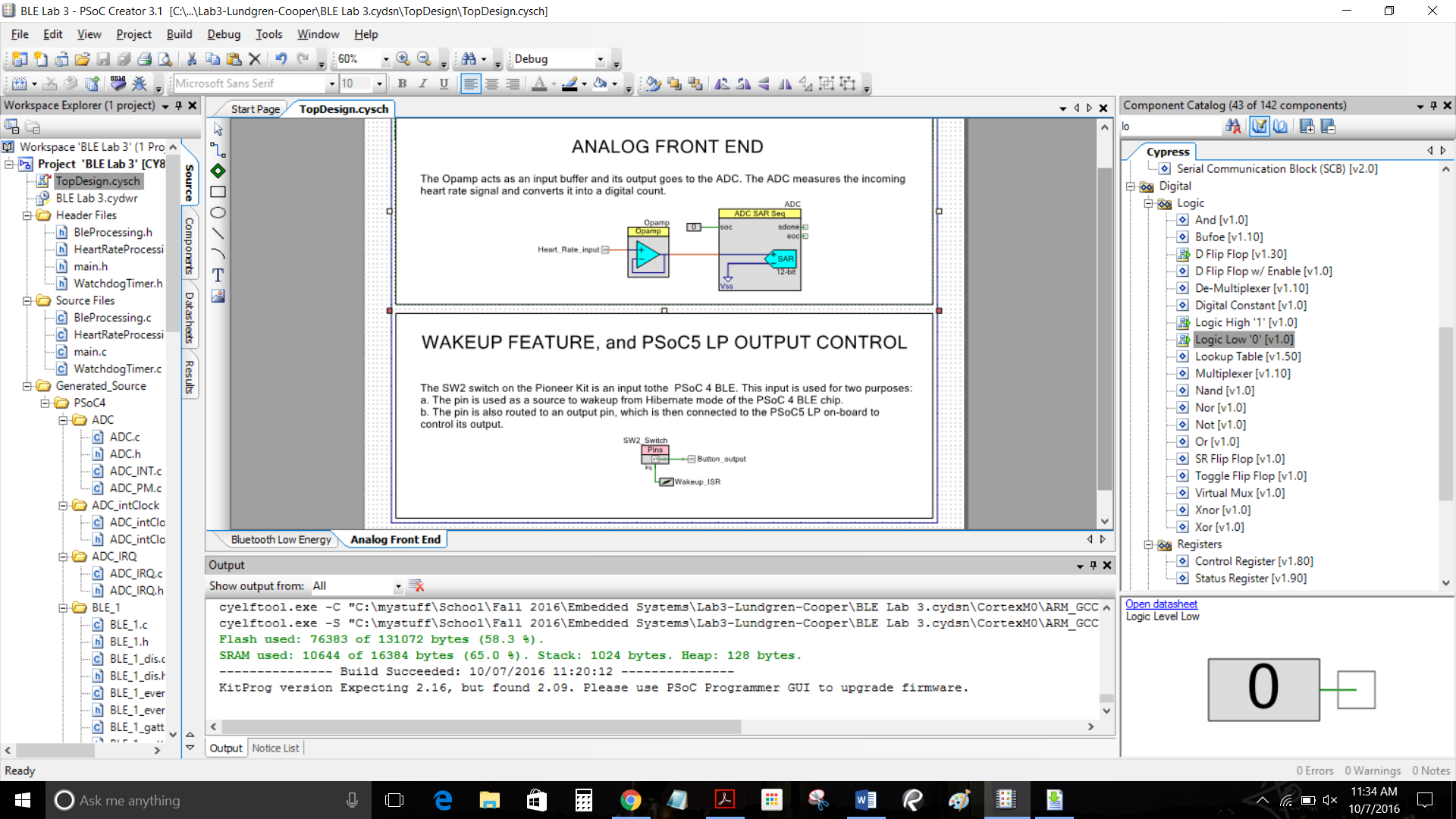


Figure 2

Figure 3 shows a successful build of the project and programing of the custom firmware to the Pioneer Kit. Using the bootloader we programed custom firmware to use the PSoC 5 as a simulation of heart rate. Holding down SW2 on the Pioneer kit puts it in bootloader mode.

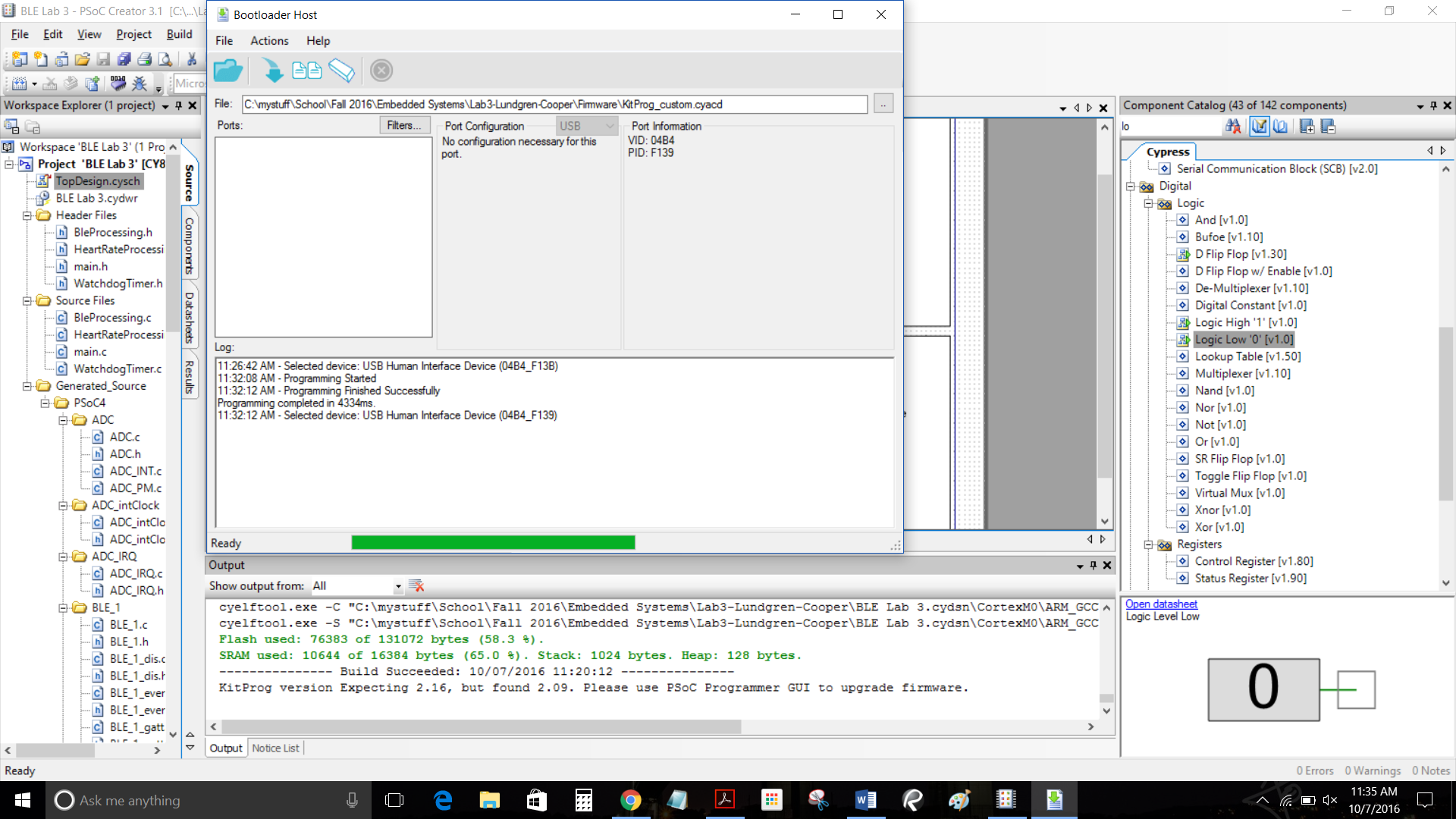


Figure 3

Next, we tested the Pioneer Kit as a hear rate monitor with CySmart and the BLE-USB Bridge. After connecting to the kit through CySmart we were able to see the value of the heart rate monitor change by pressing the SW2 button. The hex values in the Heart Rate attribute were observed changing upon pressing the SW2 button. Pressing the SW2 button was a way to simulate a pulse in a human heart rate and send that data to the kit. Then we disconnected the USB dongle to use a more practical display.

CySmart has already developed an iOS and Android application for the use with a heart rate sensor. We used the Android application to connect our kit to and view the simulated heart rate. Using the CySmart heart rate application we were able to connect to our kit. By pressing the SW2 button on the kit we were able to successfully see the changes in hear rate measured on the cell phone screen (figure 4).

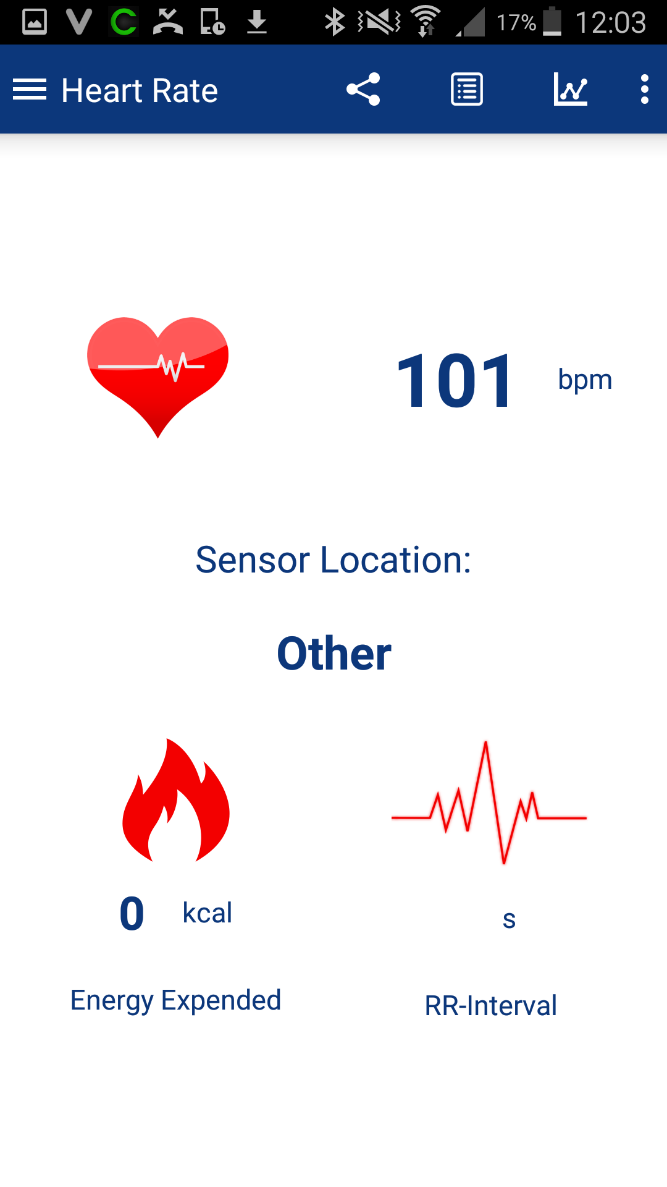
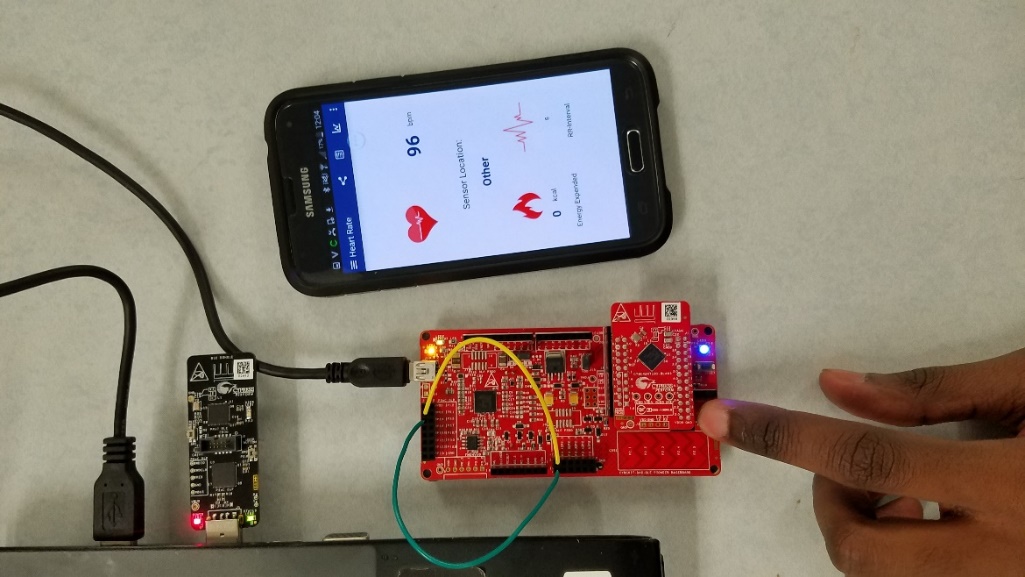


Figure 4

Additional Exercises

1. We configured the Opamp to function as a follower by double clicking the Opamp on the Workspace (figure 5). Then, in the General tab under the Deep Sleep Option select enable to use at Reduced Power and Performance. Also, the mode of the Opamp is set as a follower as to provide negative feedback and to work in Deep-Sleep mode with lower power settings.

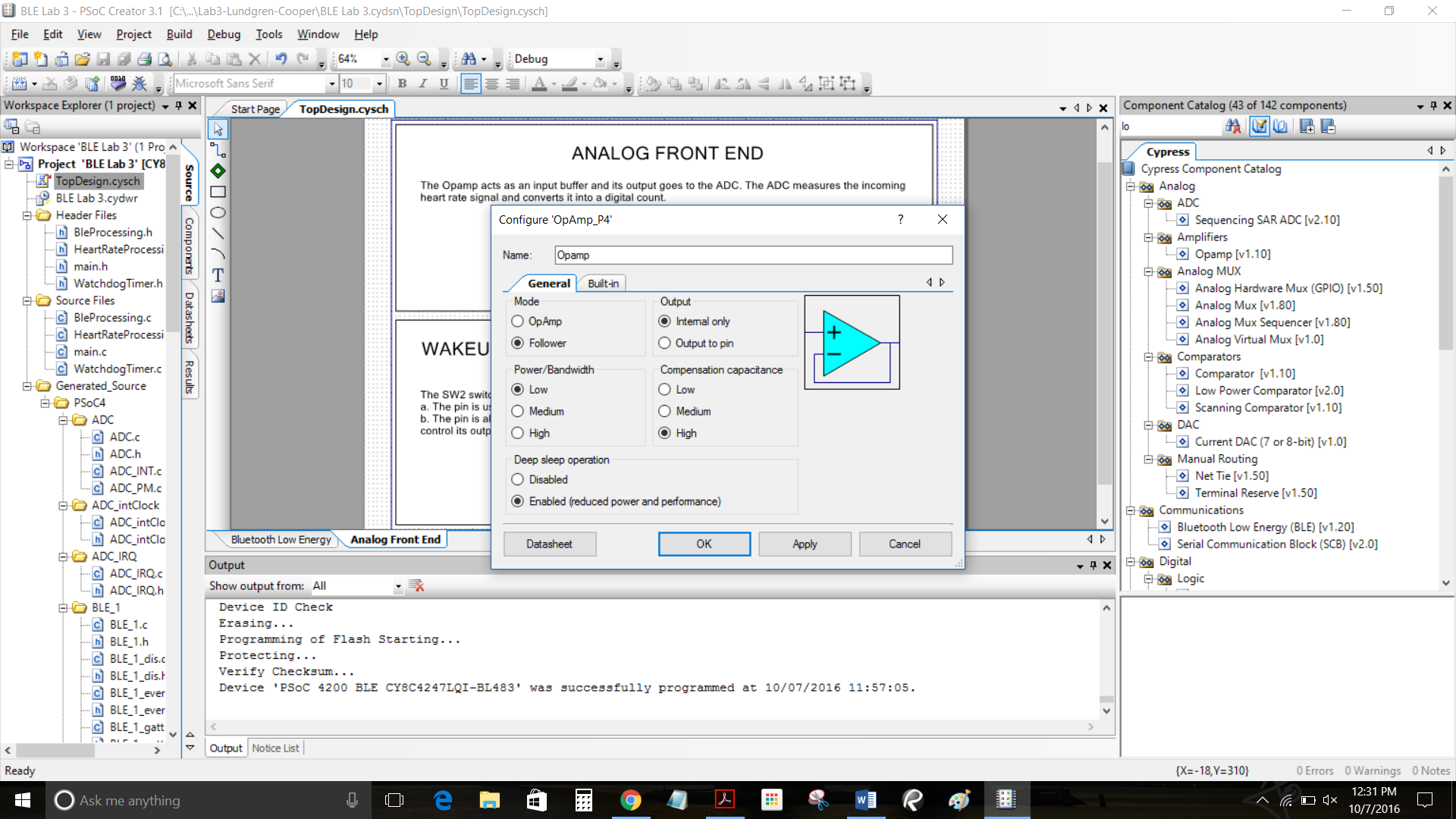


Figure 5

1. In order to change the connection interval to 1 second for the PSoC 4 BLE device we needed to change some code in the main.c file which is where the firmware is located. The API function CyBle\_L2capLeConnectionParamUpdateRequest() depends on the variable TIME\_SINCE\_CONNECTED\_MS. Changing the constant variable:

TIME\_SINCE\_CONNECTED\_MS (5000), an interval of 5 milliseconds to

TIME\_SINCE\_CONNECTED\_MS (1000000), to an interval of 1 second allowed us to change the connection time to the PSoC 4 BLE device (figure 6).

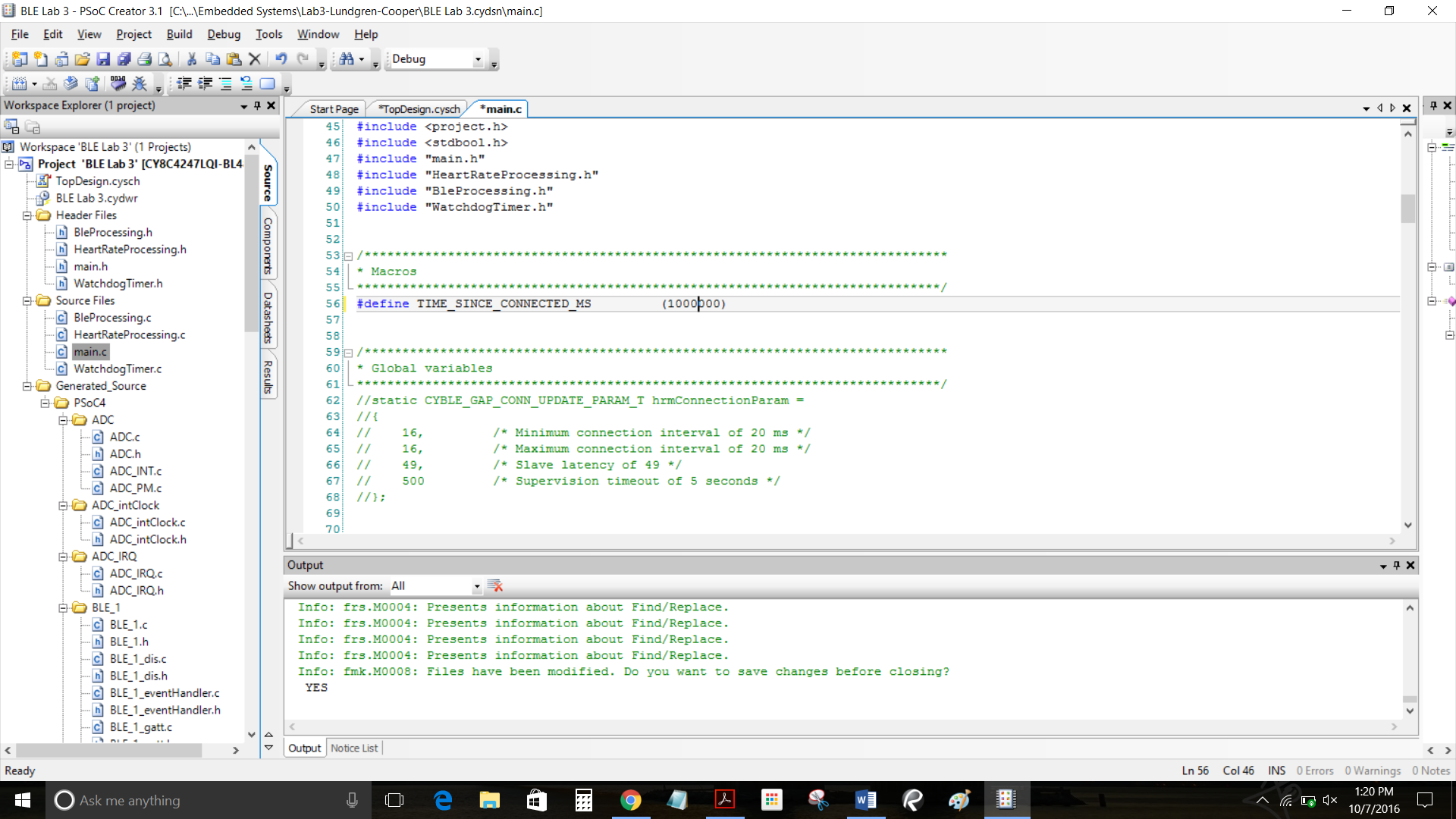


Figure 6

1. In order to drive the PSoC kit with a PWM we needed to add the component along with a clock to the Workspace. The purpose of using a PWM is to simulate a heart rate based on a clock frequency as input instead of physically pressing the SW2 switch on the kit at an interval. See figure 7 below for the addition of these components.

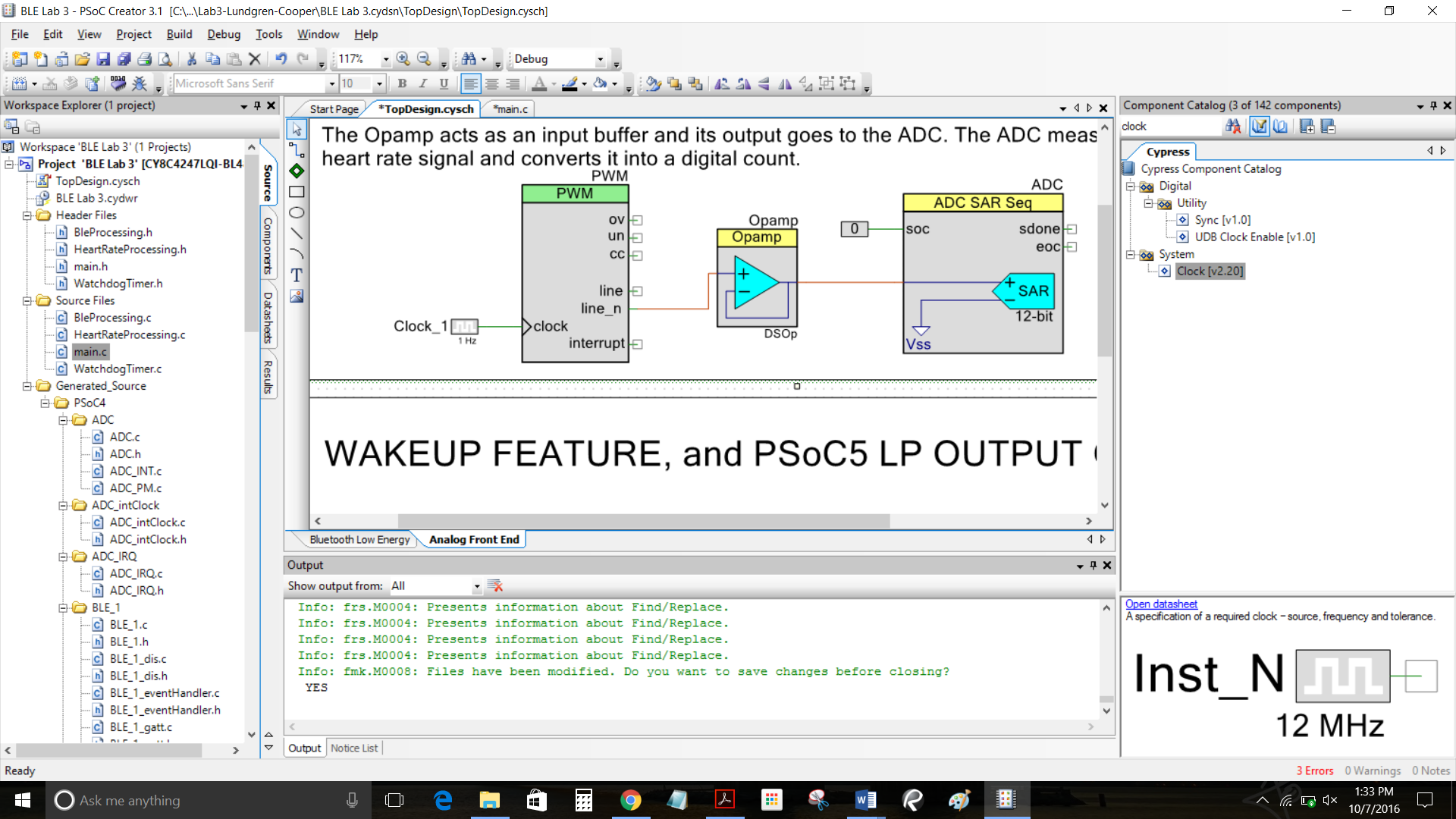


Figure 7

Then, we needed to initialize the PWM in our main.c file in order for the component to function. This was done by calling the function:

PWM\_Start();, inside the main body of the code (figure 8):

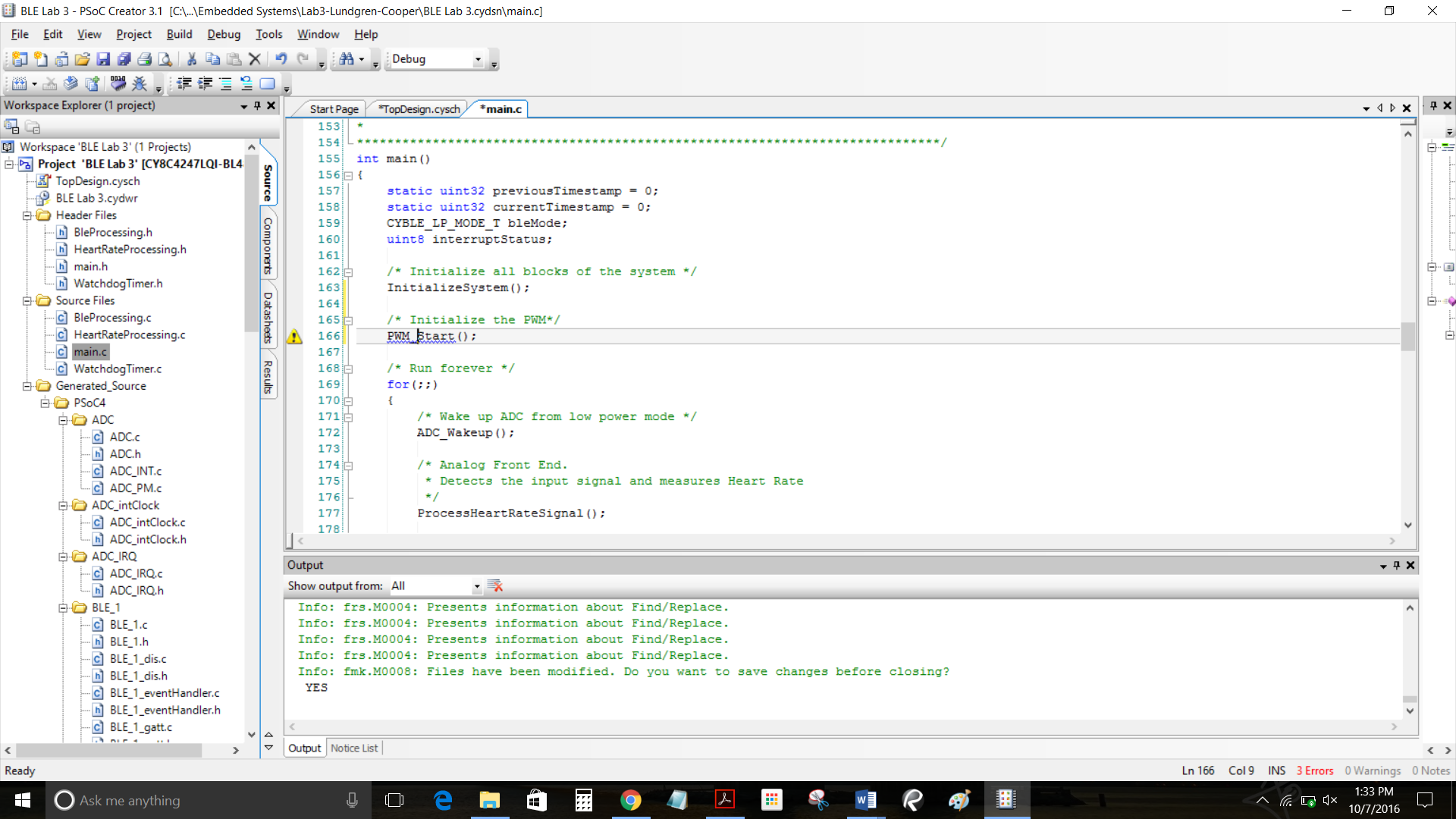


Figure 8

Lastly, because the TCPWM block is not active during the Deep-Sleep mode, we commented out the variable and function call (figure 9):

bleMode = CyBle\_EnterLPM(CYBLE\_BLESS\_DEEPSLEEP;

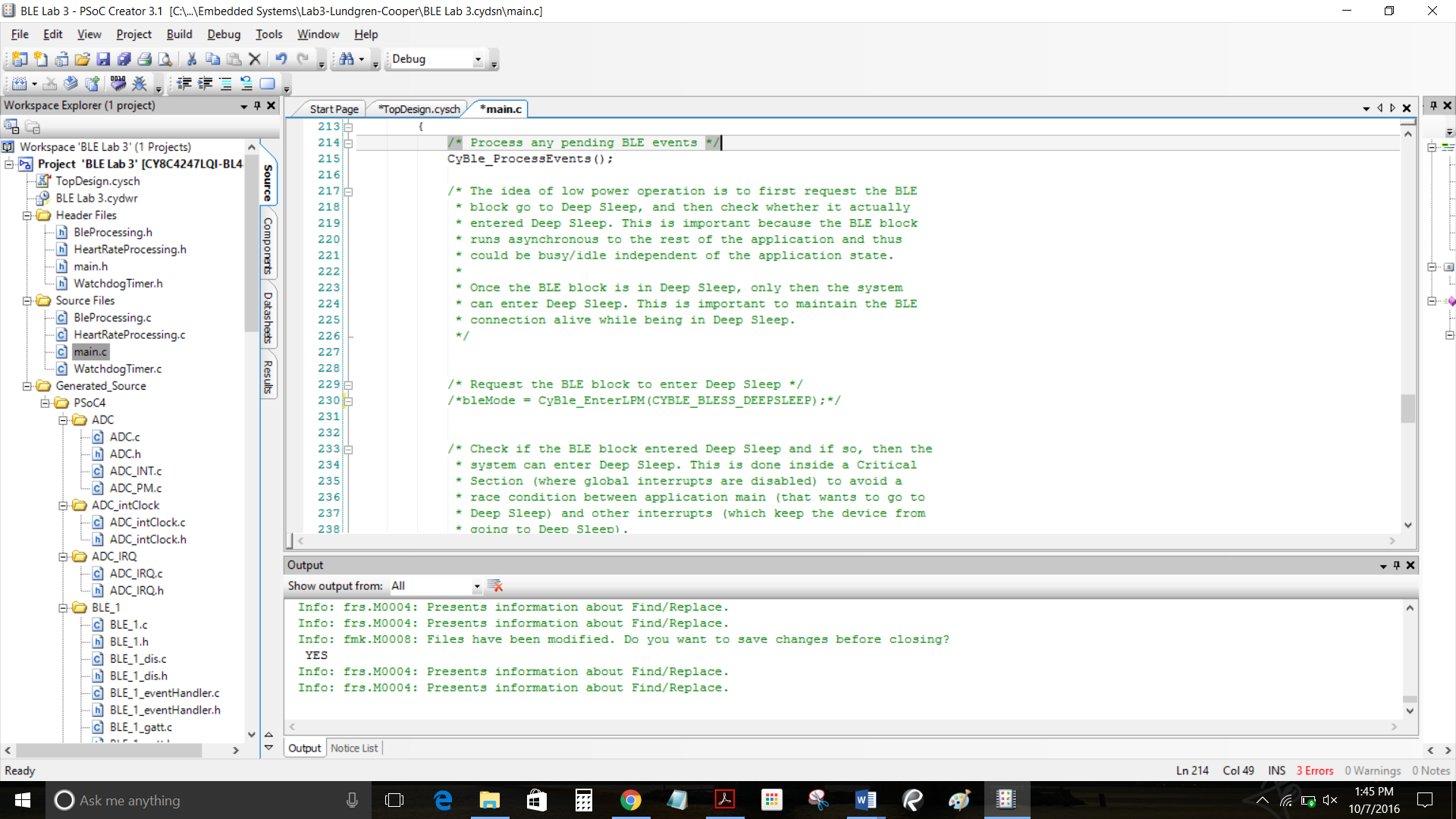


Figure 9

Conclusions

This lab was a way for us to use the Pioneer Kit and simulate the PSoC 4 as a heart rate monitor. The initial setup included a Digital Input Pin which was assigned to the SW2 button on the kit. By pressing this button, it simulated a pulse in a human heart rate. We also implemented the Deep-Sleep power mode cycle of the PSoC 4 that is in hibernation when the device is not active.

Using the Android CySmart application we were able to see the results of our configuration and simulated input. By varying the rate at which we pressed SW2 button, different hear rate values were displayed on the Android application. Next, we were able to use a PWM to simulate the input of a heart beat instead of pressing a button on the kit repeatedly.

Github link:

https://github.com/jlundgren01/Lab3.git