

LAB 4 REPORT

CAPSENSE DESIGN

WITH BLE CONNECTIVITY

Director: Dr. Ahmed Rafaey Hussein

Group Members: Mohammadamin Saburruhmonfared

Chaoao Shi / 250871446

June 23, 2016

OBJECTIVES

The main purpose of this lab is to

1. Adjust RGB LED color and intensity using the PRISM Component
2. Implement a custom BLE Profile with a custom Service to send RGB LED color and intensity over BLE
3. Implement a Custom Service to send CapSense slider data over BLE
4. Use the CySmart tool or mobile app to validate the operation

PROCEDURE

1. Block diagram

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

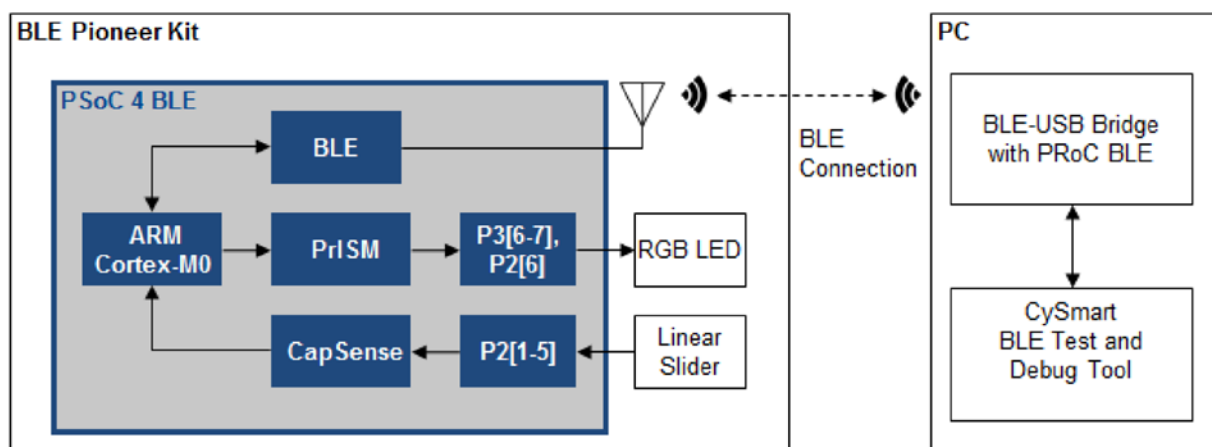


Figure 1: Block Diagram for Lab 4

2. Schematic

Open the BLE Lab 4 template provided by Cypress, and open TopDesign.cysch. In the Bluetooth Low Energy sheet of the schematic, we will place the BLE component, CapSense and RGB LED Control. In the following, we show the component configuration step by step. As shown in Figure

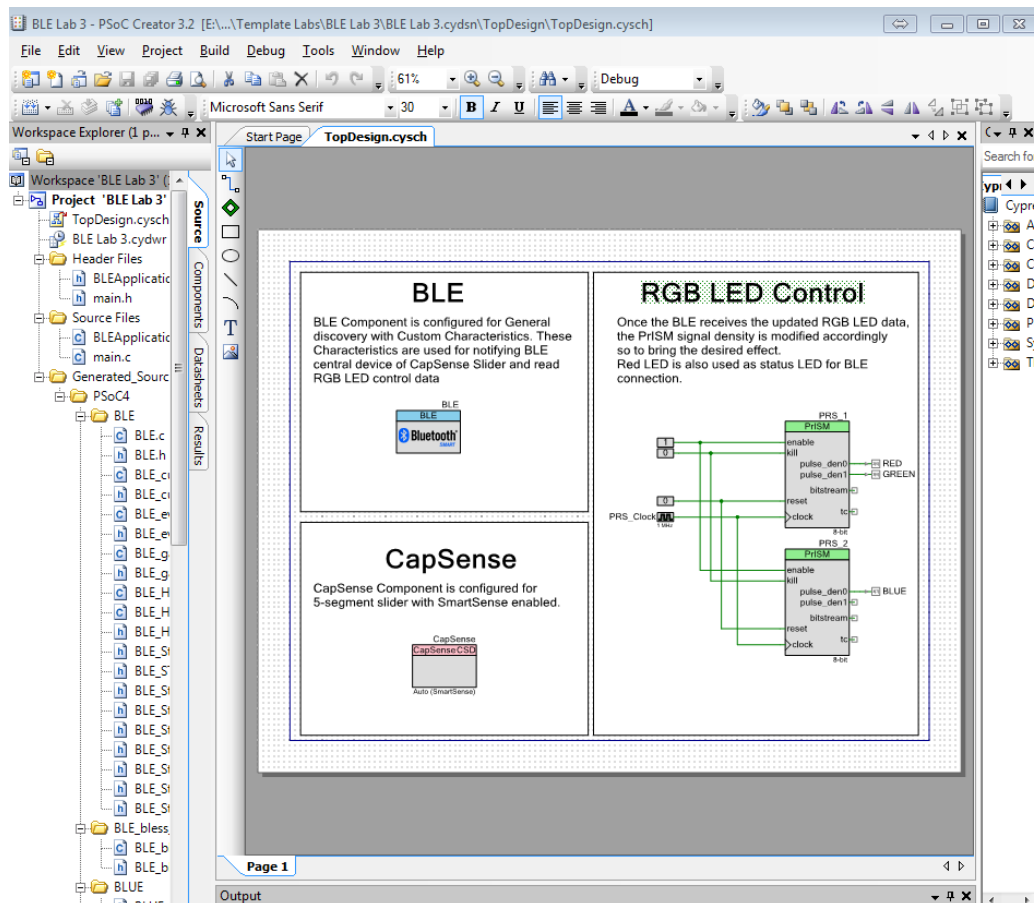


Figure 2: Schematic

3. BLE configuration

Configure the Component as shown in the following figures.

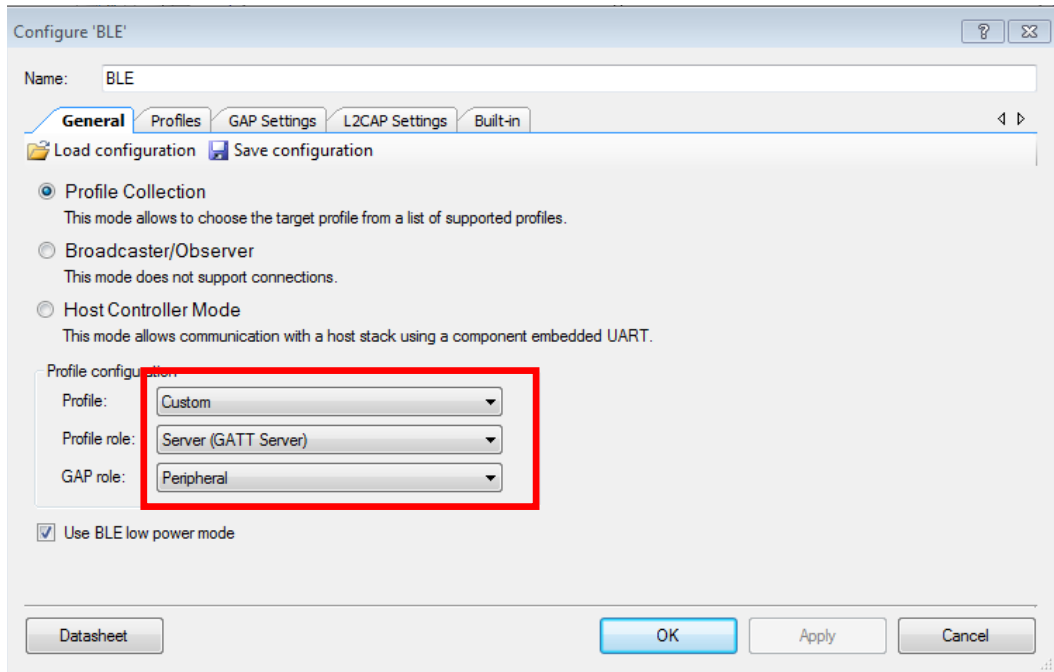


Figure 3: BLE Component Configuration – General Tab

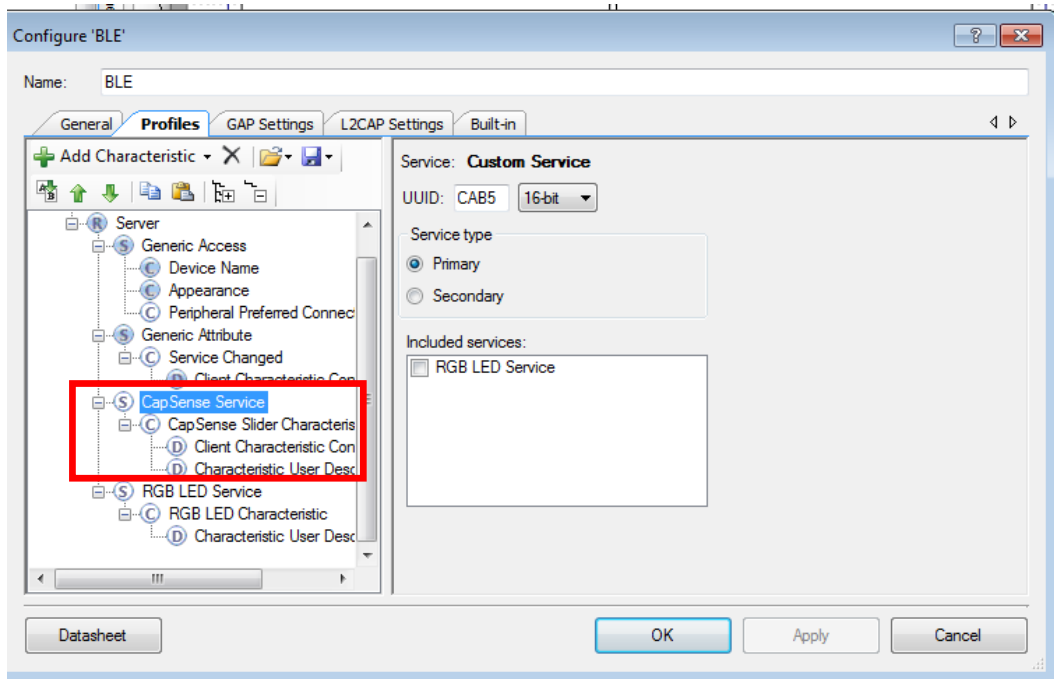


Figure 4: BLE Component Configuration – Profile Tab (Configuring Custom Service UUID for CapSense Slider)

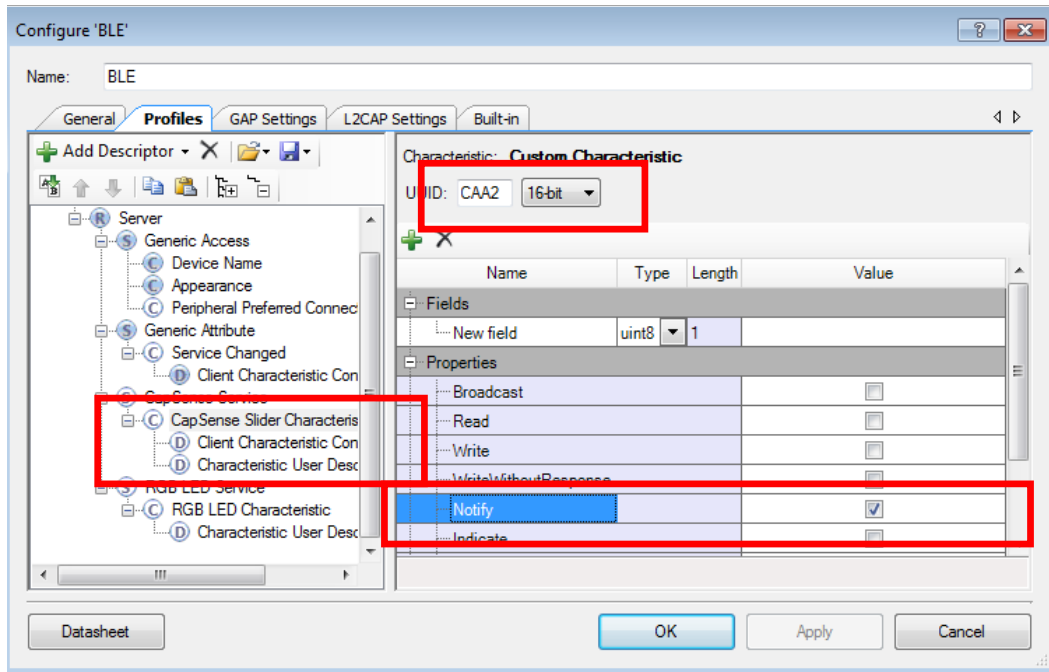


Figure 5: BLE Component Configuration – Profile Tab (Configuring Custom Characteristic for CapSense Slider)

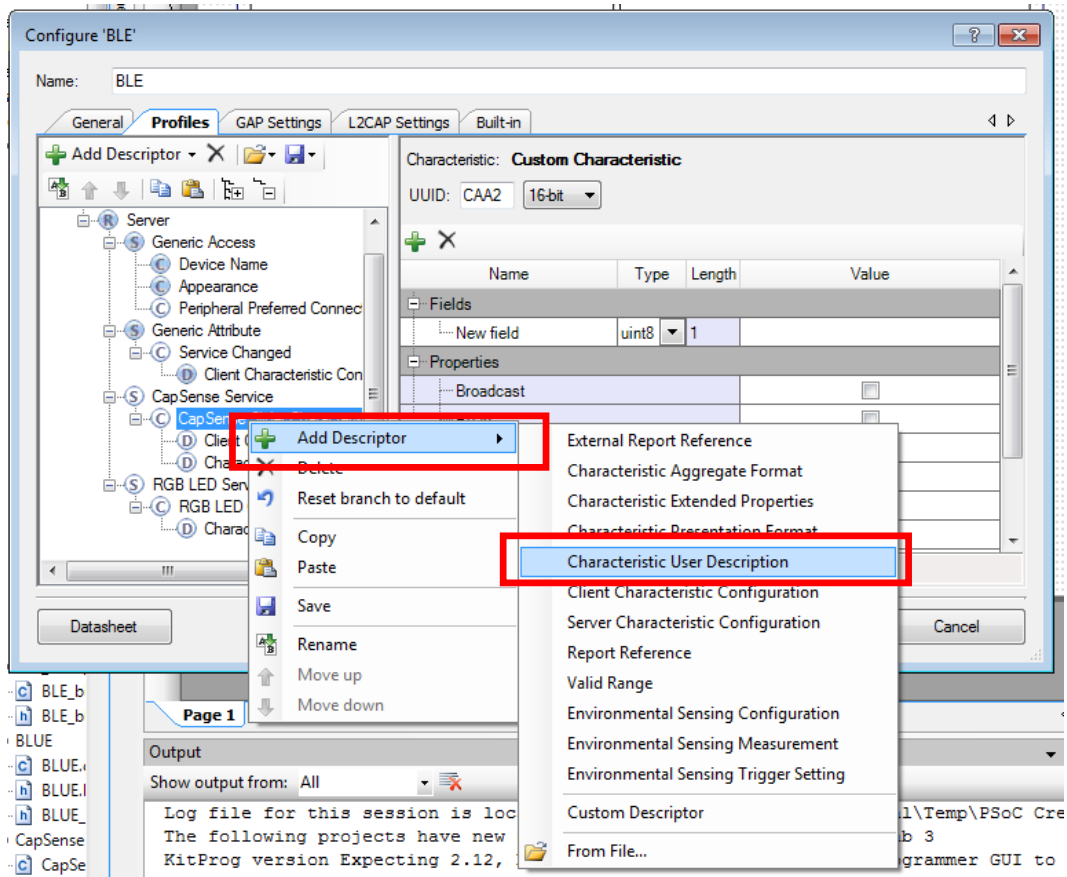


Figure 6: BLE Component Configuration – Profile Tab (Add Descriptor for the CapSense Service)

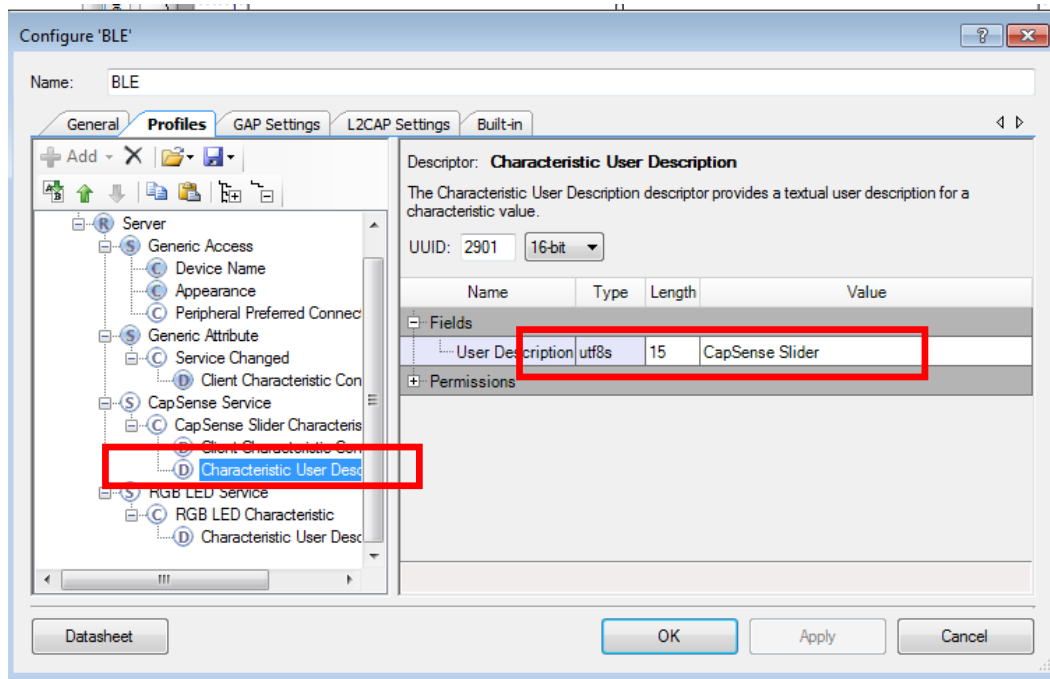


Figure 7: BLE Component Configuration – Profile Tab (Characteristic User Description for CapSense Slider Characteristic)

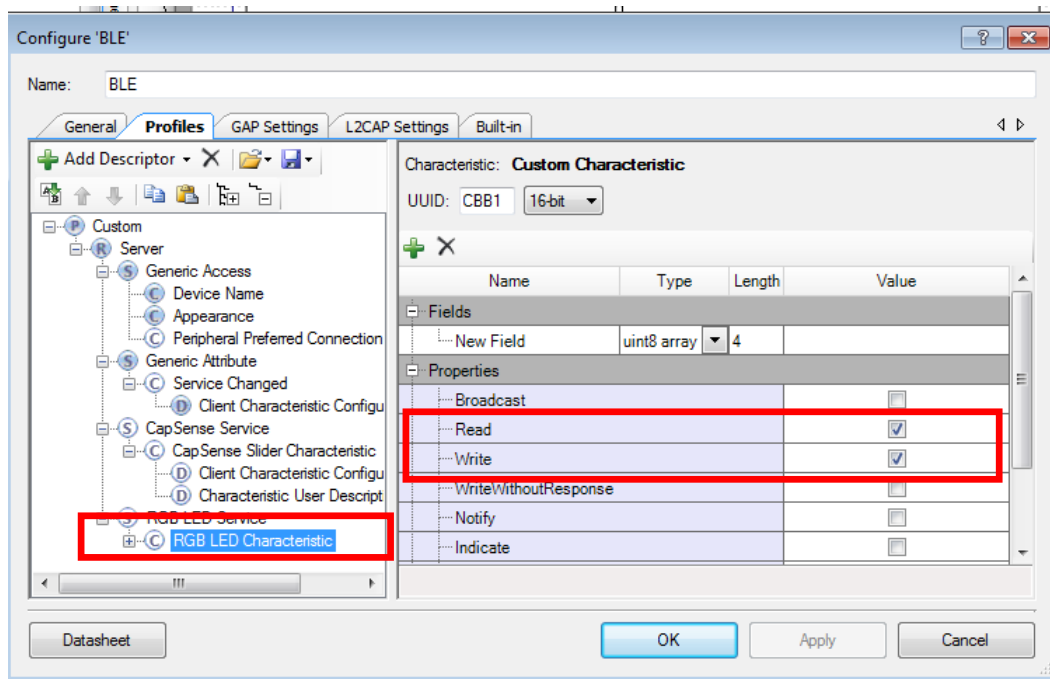


Figure 8: BLE Component Configuration – Profile Tab (Configuring Characteristic for RGB LED Service)

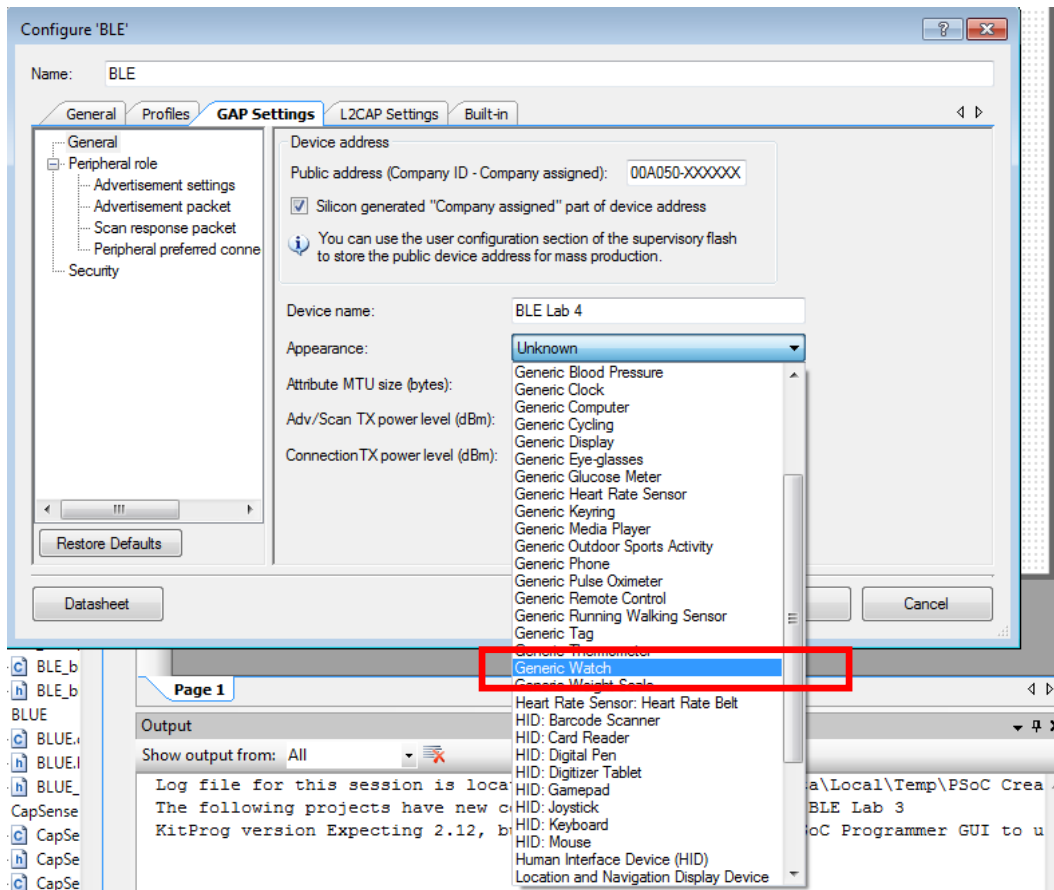


Figure 9: BLE Component Configuration – Gap Setting Tab

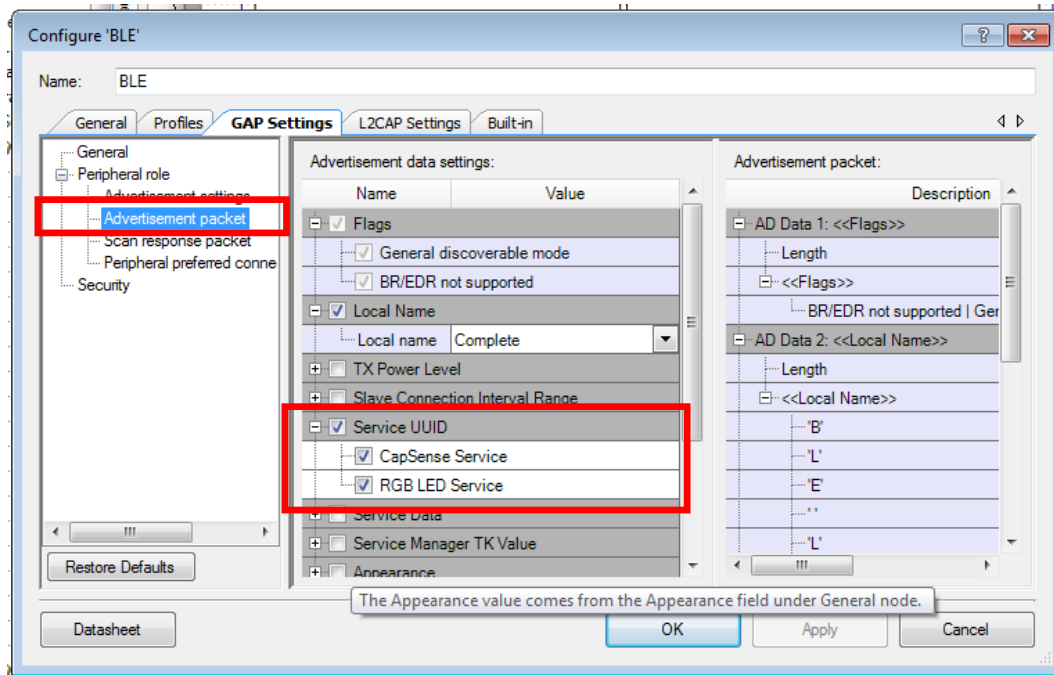


Figure 10: BLE Component Configuration – GAP Settings (Advertisement Packets)

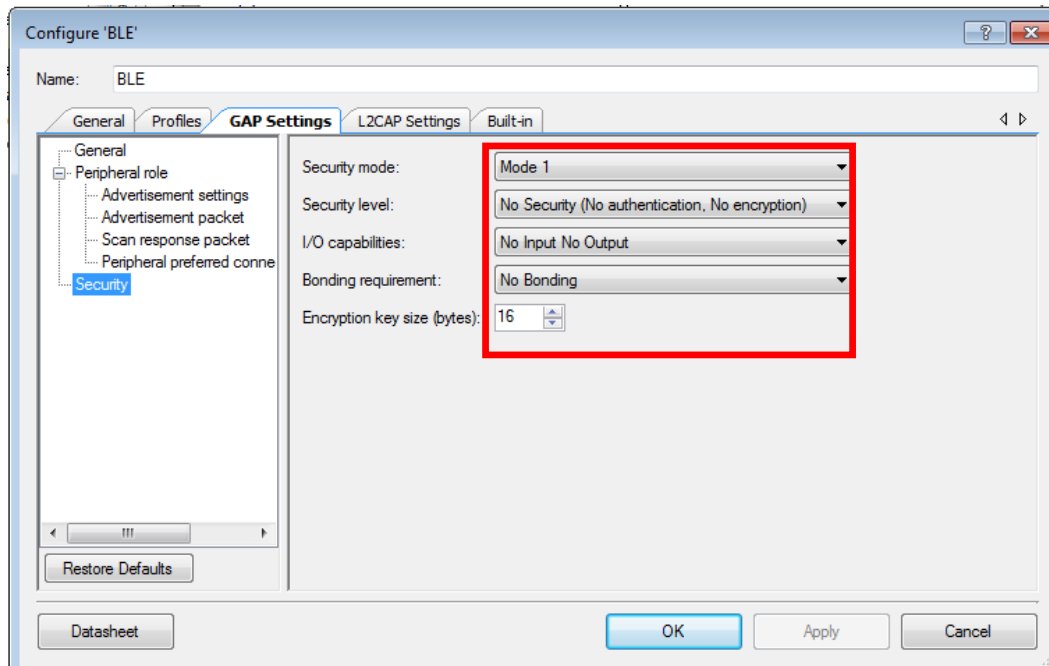


Figure 11: BLE Component Configuration – GAP Settings-Security

4. CapSense CSD configuration

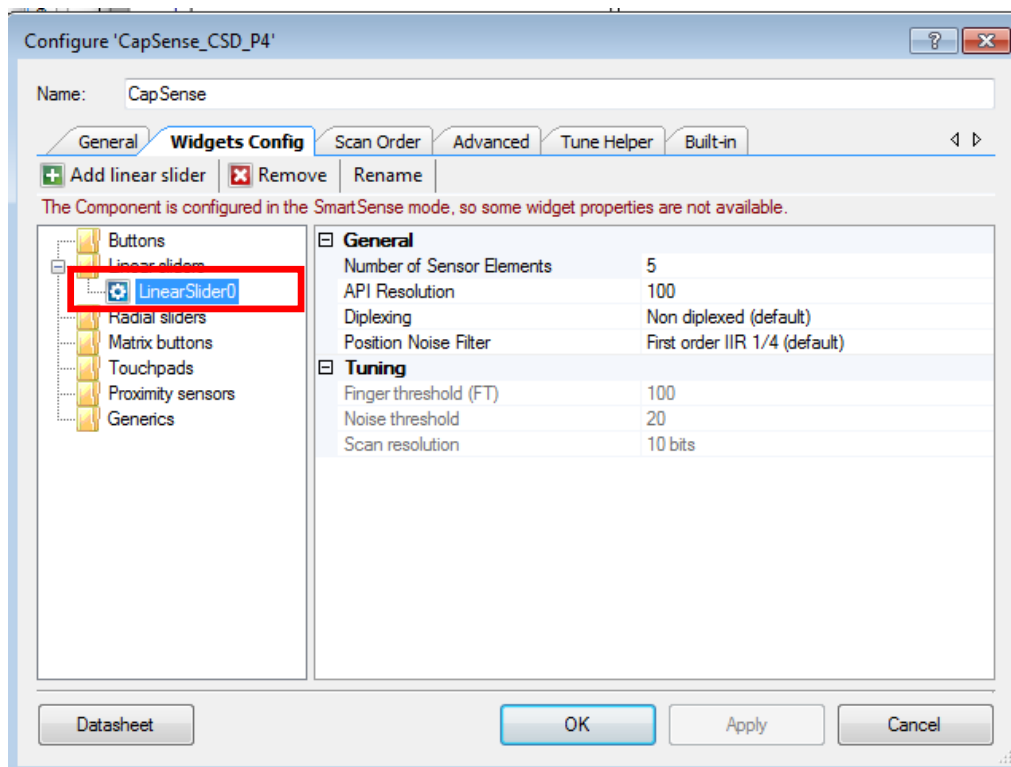


Figure 12: CapSense CSD Component Configuration

5. DWR configuration

Now we are done with configuring the schematics of the components. We are going to configure the DWR file, which is a Design Wide Resources. The results are shown in Figure 13.

| Alias | Name | Port |
|---------------------|-------------------|--------------------------------------------------------------------------------|
| Cmod | \CapSense:Cmod\ | P4[0] CSD:c_mod, TCPWM0:line_out, SCB1:uart rts, SCB1:spi mosi |
| LinearSlider0_e0_LS | \CapSense:Sns[0]\ | P2[1] OA0:vminus, SCB0:spi_select[2] |
| LinearSlider0_e1_LS | \CapSense:Sns[1]\ | P2[2] OA0:vout_10x, SRSS:wakeup, SCB0:spi_select[3] |
| LinearSlider0_e2_LS | \CapSense:Sns[2]\ | P2[3] OA1:vout_10x, SRSS:ext_clk_lf |
| LinearSlider0_e3_LS | \CapSense:Sns[3]\ | P2[4] OA1:vminus |
| LinearSlider0_e4_LS | \CapSense:Sns[4]\ | P2[5] OA1:vplus |
| | BLUE | P3[7] SARMUX:pads[7], TCPWM3:line_out_compl, SCB1:uart cts, SRSS:ext_clk_lf |
| | GREEN | P3[6] SARMUX:pads[6], TCPWM3:line_out, SCB1:uart rts |
| | RED | P2[6] OA0:vplus_alt |

Figure 13: Pins Assignment

6. Build and Program

```

The link step is up to date, no work needs to be done.
===== Build Succeeded: 06/23/2016 16:16:02 =====
===== Build Started: 06/23/2016 16:11:15 Project: BLE Lab 3, Configuration: ARM GCC 4.8.4 Debug =====
The code generation step is up to date.
arm-none-eabi-gcc.exe -mcpu=cortex-m0 -mthumb -Wno-main -I. -IGenerated_Source\PSoC4 -Wa,-ahb*.CortexM0\ARM_GCC_484\Debug\main.lst -g -D DEBUG -Wall -ffunction-sections -O0 -c *.main.c -o *.main.o
arm-none-eabi-ar.exe -rc *.CortexM0\ARM_GCC_484\Debug\BLE Lab 3.a* *.CortexM0\ARM_GCC_484\Debug\cyfitter_cfg.o *.CortexM0\ARM_GCC_484\Debug\GREEN.o *.CortexM0\ARM_GCC_484\Debug\RED.o *.CortexM0\ARM_GCC_484\Debug\BLUE.o
arm-none-eabi-objcopy.exe -O binary *.CortexM0\ARM_GCC_484\Debug\BLE Lab 3.o *.CortexM0\ARM_GCC_484\Debug\BLE Lab 3.o
arm-none-eabi-gcc.exe -Wl,--start-group -o *.CortexM0\ARM_GCC_484\Debug\BLE Lab 3.elf* *.CortexM0\ARM_GCC_484\Debug\main.o *.CortexM0\ARM_GCC_484\Debug\BLEApplications.o *.CortexM0\ARM_GCC_484\Debug\cyfitter.o *.CortexM0\ARM_GCC_484\Debug\GREEN.o *.CortexM0\ARM_GCC_484\Debug\RED.o *.CortexM0\ARM_GCC_484\Debug\BLUE.o
arm-none-eabi-objcopy.exe -O binary *.CortexM0\ARM_GCC_484\Debug\BLE Lab 3.elf* *.CortexM0\ARM_GCC_484\Debug\BLE Lab 3.o
cyfload.exe -c "E:\2016.05.27_BLE\Cypress BLE Workshop\Workshop Exercises and Manuals\Template Labs\BLE Lab 3\BLE Lab 3.cydsn\CortexM0\ARM_GCC_484\Debug\BLE Lab 3.elf" --flash_row_size 128
cyfload.exe -c "E:\2016.05.27_BLE\Cypress BLE Workshop\Workshop Exercises and Manuals\Template Labs\BLE Lab 3\BLE Lab 3.cydsn\CortexM0\ARM_GCC_484\Debug\BLE Lab 3.elf"

===== Build Succeeded: 06/23/2016 16:11:17 =====
Programming started for device: 'PSoC 4200 BLE CY8C4247IQT-BL493'.
=====

Erasing...
Programming of Flash Starting...
Protecting...
Verify Checksum...
Device 'PSoC 4200 BLE CY8C4247IQT-BL493' was successfully programmed at 06/23/2016 16:11:27.

```

Figure 14: Successful Programmed to Kit

7. RESULTS

1. Testing via 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.

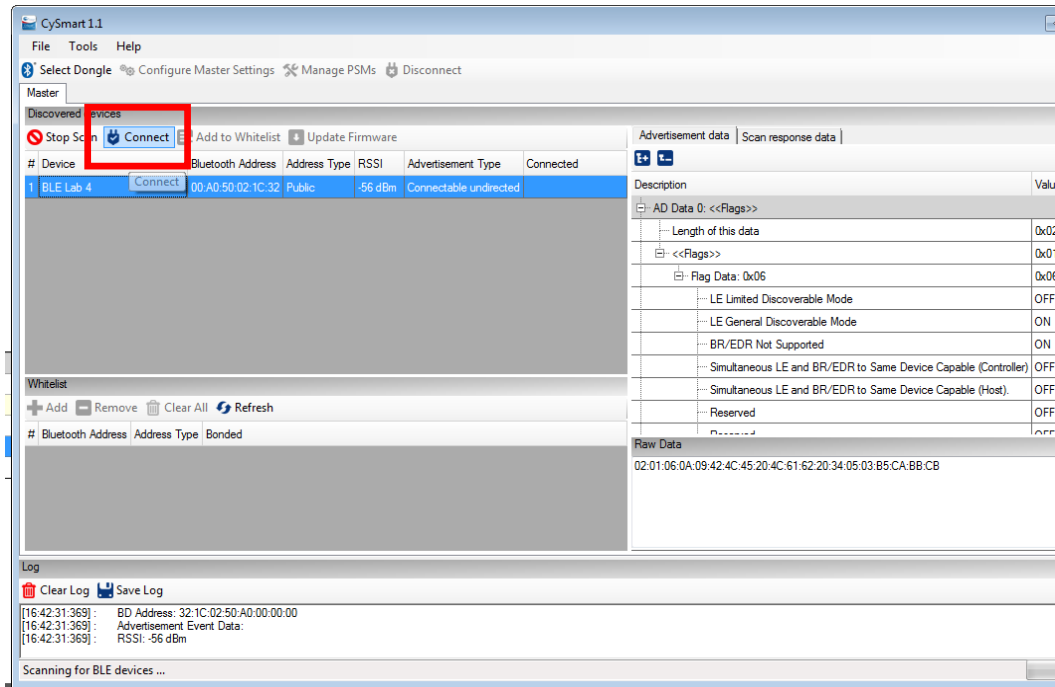


Figure 15: CySmart Scanning Results

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 16. Then Write a 4-byte value to this Characteristic on the right. As shown Fig 17. This comment makes the LED on the board, blue.

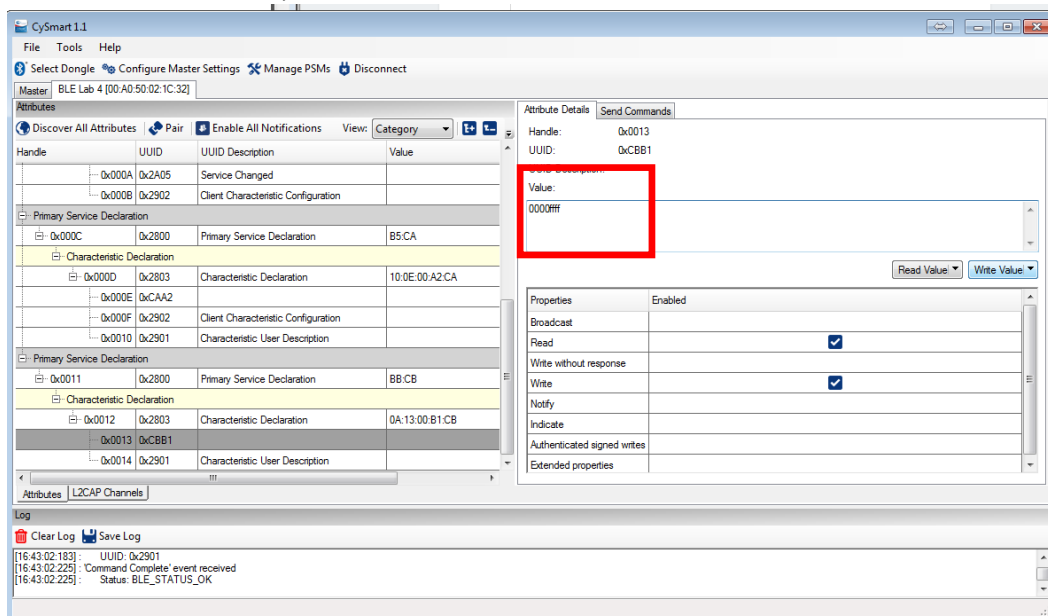


Figure 16: CySmart - Slider Notifications

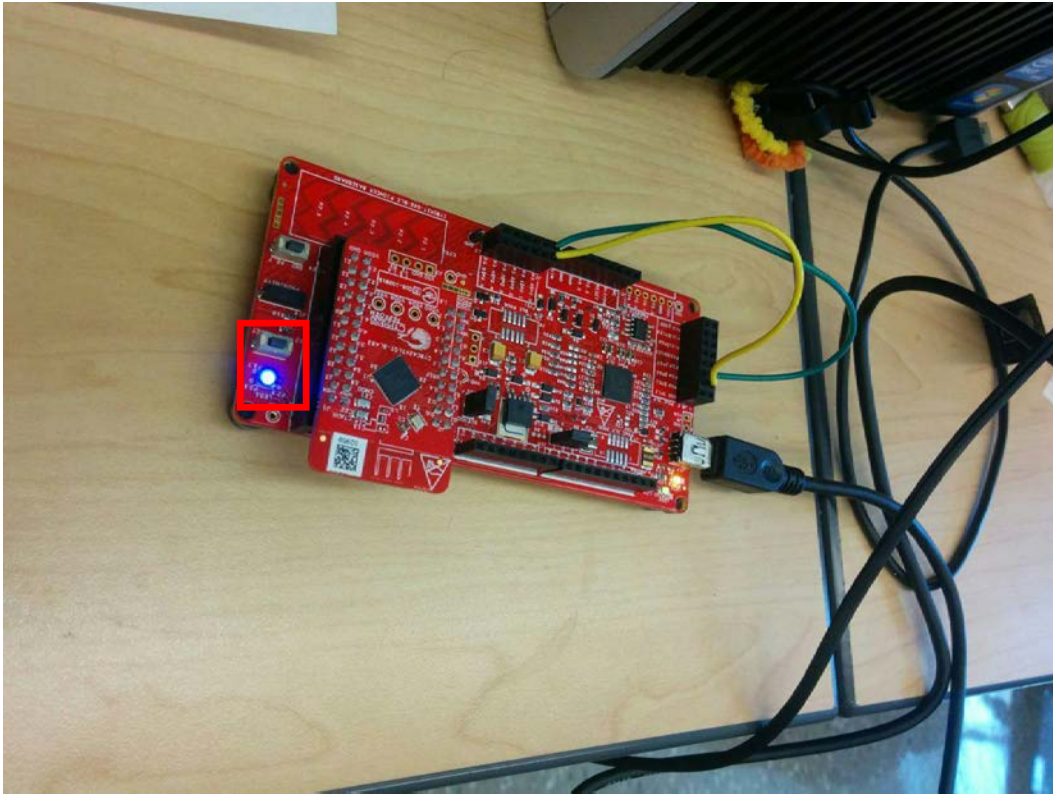


Figure 17: Result demonstration

As Shown in Fig 18, by Moving finger over the slider on the kit, the value of the Characteristic changes in the CySmart tool.

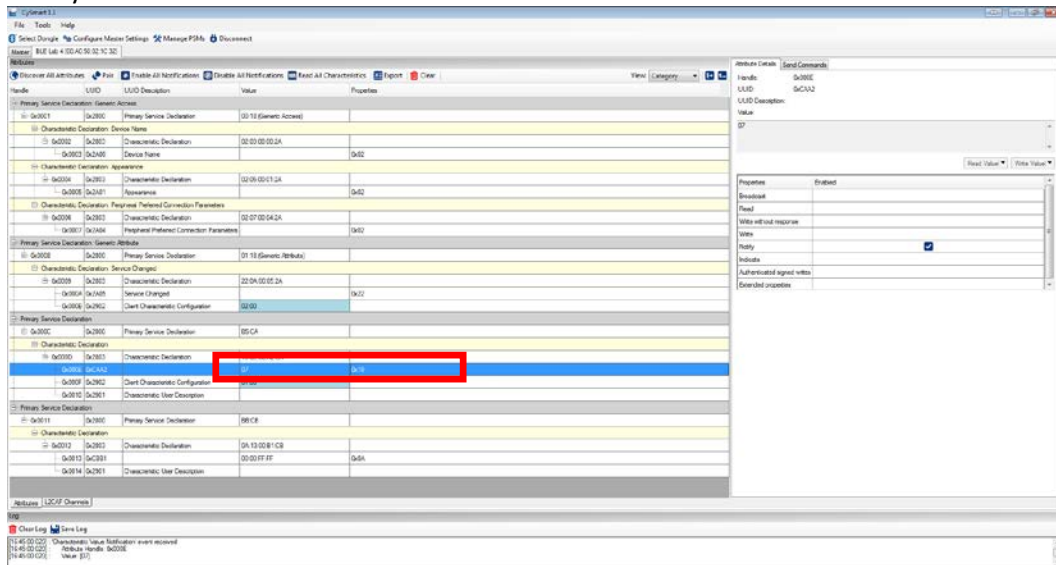


Figure 18: CySmart - Slider Notifications

2. Testing via android application

We also tested the project by android application. The results shown in Fig 19 and Fig 20.

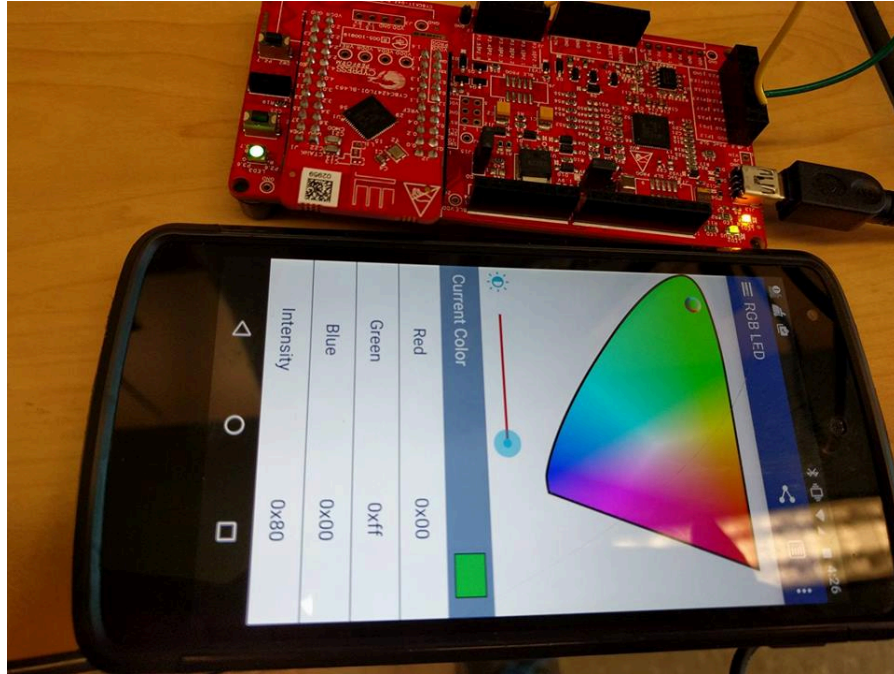


Figure 19: Result demonstration

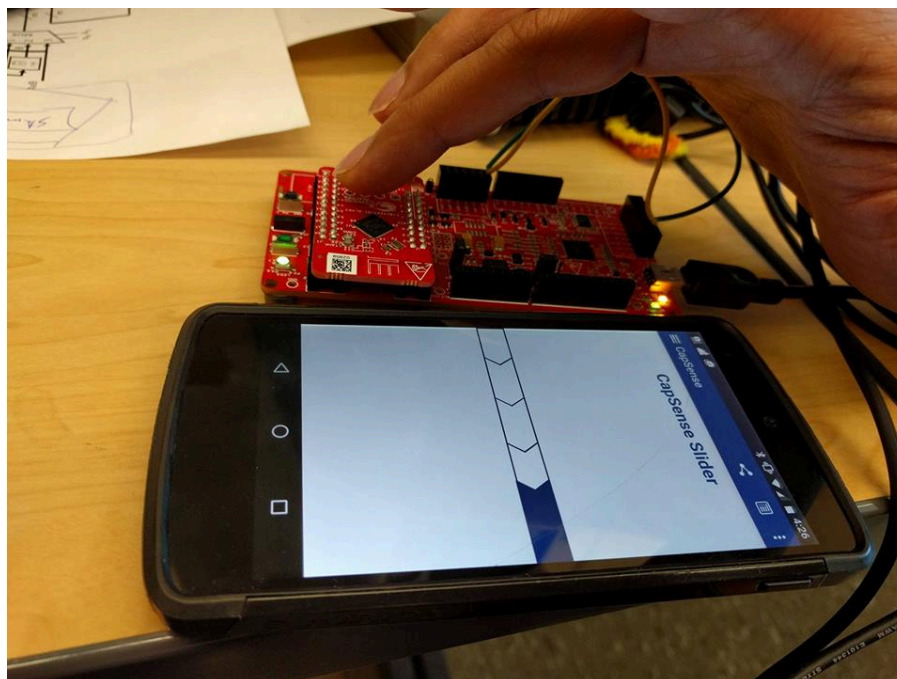


Figure 20: Result demonstration

CONCLUSIONS

By completing Lab 4, we have learned how to create a Custom Profile by implementing an RGB LED controller through BLE as well as how to combine CapSense and BLE in a system, by designing a slider application.