BLE / WIFI Lab

For the MAX® 10 DECA FPGA Evaluation Kit

Version 15.0 **5/15/2015**

TABLE OF CONTENTS

LAB 7. INTERACT WITH DECA USING BLE AND WI-FI 270
7.1 **System Overview and Architecture** 270
7.1.1 Wi-Fi Subsystem 270
7.1.2 Bluetooth® Subsystem 271
7.2 **Set up the DECA with the BLE/Wi-Fi Cape** 272
7.2.1 Attach the BLE/Wi-Fi cape to the DECA 272
7.2.2 Open the Quartus II Project 272
7.2.3 Download the hardware configuration file (.sof) to the MAX10 273
7.2.4 Download the software executable (.elf) to the Nios II Soft Processor 275
7.2.5 Interact with the DECA board over Wi-Fi 278
7.2.6 Interact with the DECA board over Bluetooth® LE 282

LAB 7. INTERACT WITH DECA USING BLE AND WI-FI

Overview: In this lab, you will interact with the DECA platform from your Android or iOS smartphone over both Bluetooth® and Wi-Fi using the BeagleBone-compatible Wi-Fi cape from Dallas Logic. You'll be able to visit a webpage hosted on the DECA to see sensor data and change LEDs by connecting to the DECA as a Wi-Fi internet access point. You'll also be able to use a BLE app for iOS and Android to look at this data over Bluetooth® Low Energy.

MAX 10 FPGAs revolutionize non-volatile integration by delivering advanced processing capabilities in a low-cost, instant-on, small form factor programmable logic device. The devices also include full-featured FPGA capabilities such as digital signal processing, analog functionality, Nios II embedded processor support and memory controllers.

The DECA includes a variety of peripherals connected to the FPGA device, such as 4Gb DDR3L, MIPI camera interface, 10/100 Ethernet, temperature sensor, ambient light and gesture sensors, LEDs, capsense buttons and a BeagleBone compatible header.

Before continuing with this Tutorial, ensure that the Altera tools and drivers have been installed. Please refer to Lab 1 for instructions.

7.1 System Overview and Architecture

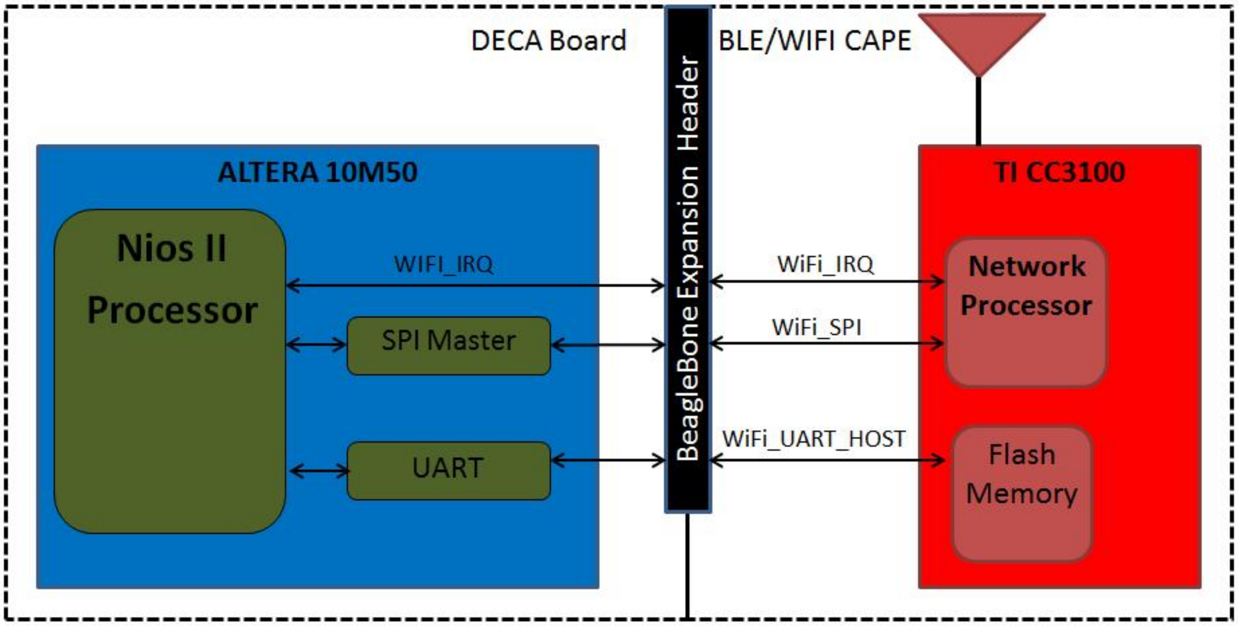
In this section, you will learn about how the MAX10 on the DECA interacts with the TI CC3100 Wi-Fi Network Processor and the CC2650 BLE chip on the BLE/Wi-Fi cape to create a wireless access point and a pairable Bluetooth® device. The Wi-Fi/BLE cape from Dallas Logic connects to the DECA board through the BeagleBone-compatible header giving the DECA board wireless capabilities.

7.1.1 Wi-Fi Subsystem

The Nios II processor communicates with the TI CC3100 Wi-Fi Network Processor over a SPI interface. The Nios II connects to a SPI master controller which manages data transfers to and from the CC3100. A parallel IO is used to implement the IRQ signal from the CC3100. Additionally, a UART interface is implemented to provide access to the on-chip flash memory on the CC3100. If the user wanted to change the website hosted on the CC3100 (as html code), the user would re-write the flash through this interface.

The CC3100 is a complete network processing solution on a single chip. The dedicated ARM MCU completely offloads Wi-Fi and Internet protocols from the host by implementing the entire TCP/IP stack, crypto and security engines, and an 802.11 b/g/n radio. The CC3100 has a dedicated on-board antenna which it uses to broadcast its wireless signal. A power management subsystem includes DC-DC converters and enables low power consumption modes such as hibernation drawing only 4 µA of power.

7.1.1.1 Below is the Wi-Fi subsystem block diagram.

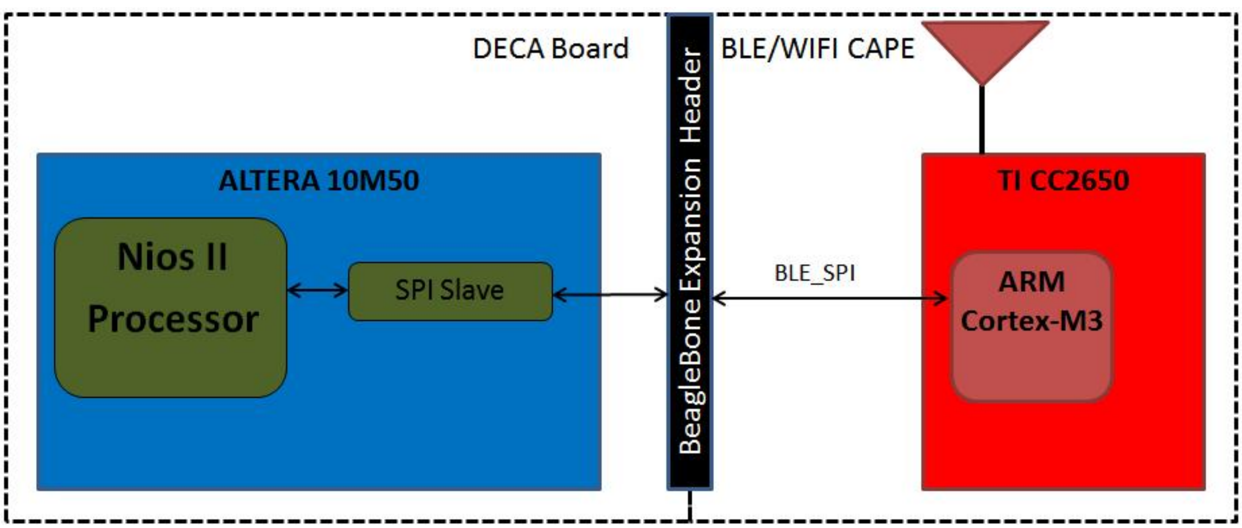


7.1.2 **Bluetooth® Subsystem**

The Nios II processor communicates with the CC2650 BLE MCU over a SPI interface. The Nios II connects to a SPI slave peripheral and receives data and requests from the CC2650 SPI master.

The CC2650 contains a 32-bit ARM Cortex-M3 processor to implement the Bluetooth® Low-Energy (Bluetooth Smart ®) stack and the 802.15.4 MAC which are both embedded into on-chip ROM. The M3 processor manages all data transfers between the connected Bluetooth device and the Nios II processor, abstracting the BLE protocol away from the Nios II host. The CC2650 has a dedicated on-board antenna to send and receive data over the Bluetooth link.

7.1.2.1 Below is the Bluetooth® subsystem block diagram.



7.2 Set up the DECA with the BLE/Wi-Fi Cape

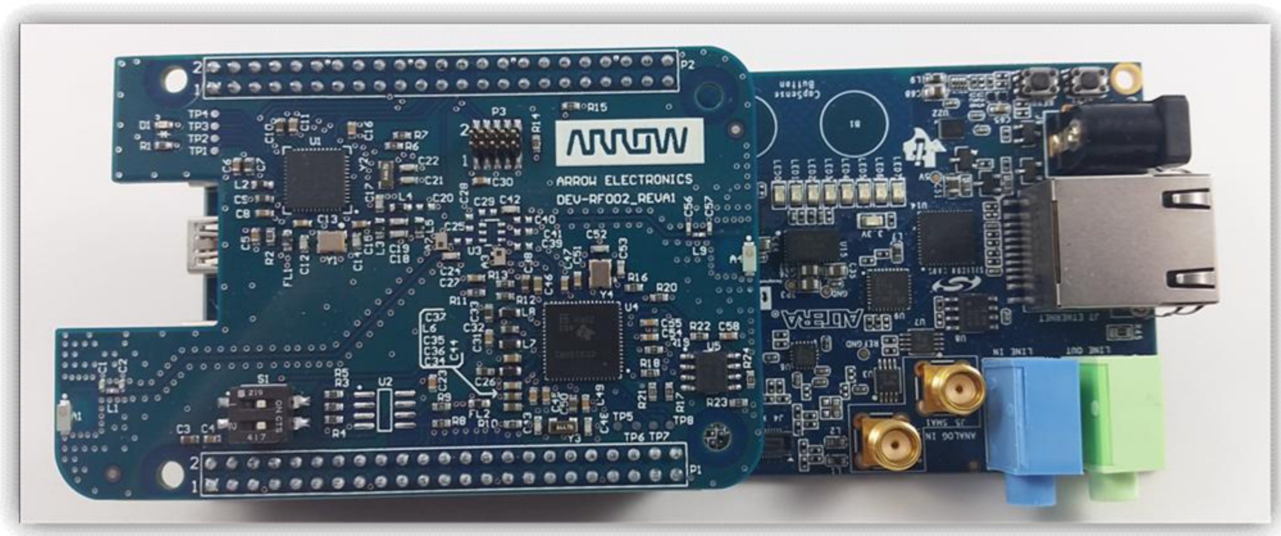
In this section, you will connect the BLE/Wi-Fi cape from Dallas Logic to the DECA board using the BeagleBone compatible header. Then you will download the hardware and software programming files to the board to get it ready to interact with.

7.2.1 Attach the BLE/Wi-Fi cape to the DECA

7.2.1.1 Place the DECA board on the table with the blue/green audio jacks pointing toward you.

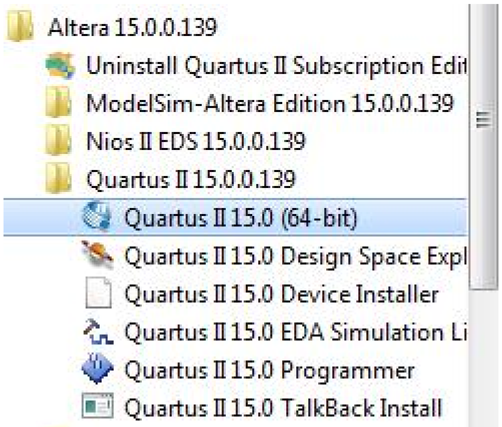
7.2.1.2 Align the BLE/Wi-Fi cape so that the Arrow logo is in the readable orientation and press the male header pins on the cape firmly into the female header socket on the DECA. The final assembly should match the image below. Ensure power is OFF.

Note: If you connect the cape incorrectly, you can permanently damage the hardware!. Ensure that the board matches the picture below



7.2.2 Open the Quartus II Project

7.2.2.1 Launch Quartus II 15.0 (64-bit) from the Start menu.



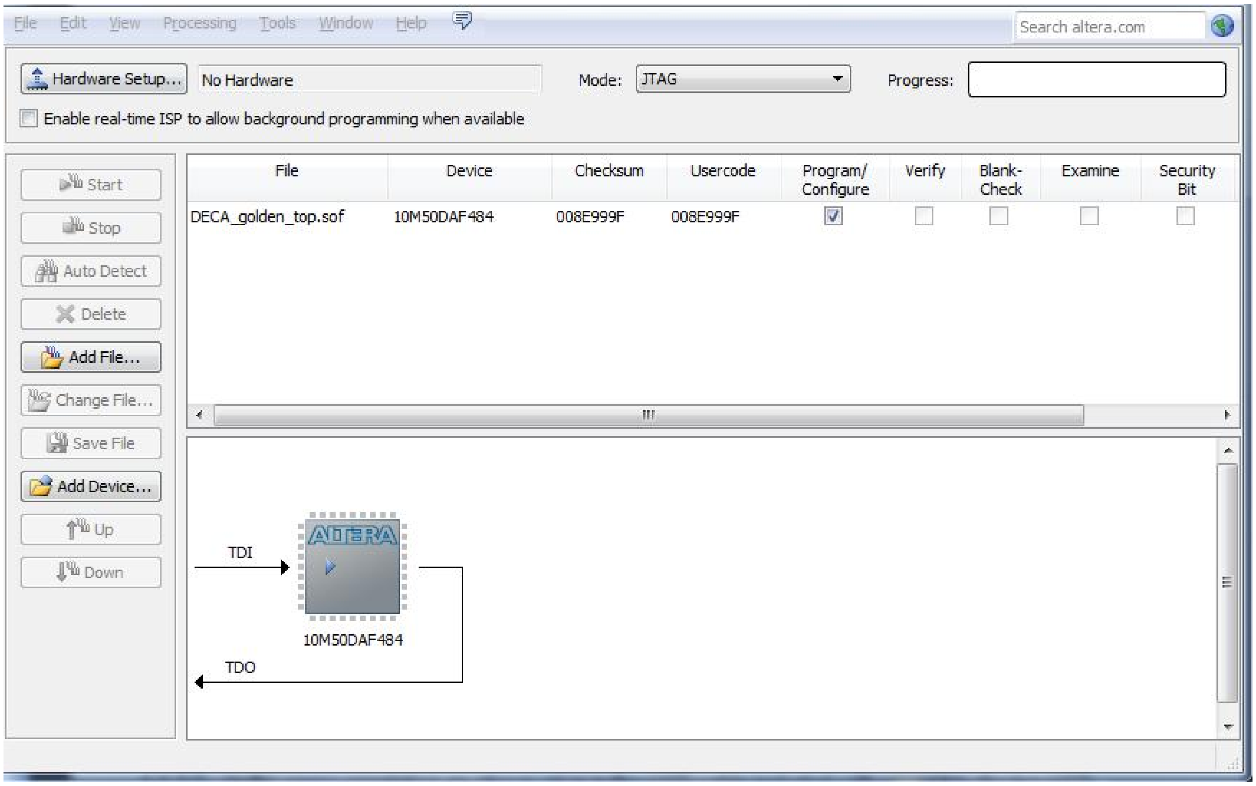
7.2.2.2 Open the workshop project from **File →Open Project**.

7.2.2.3 Browse to the directory where you extracted the lab files (for example **`C:\DECA\workshop_labs\7_BLE_WIFI_Lab`**) and open **`DECA_golden_top.qpf`**.

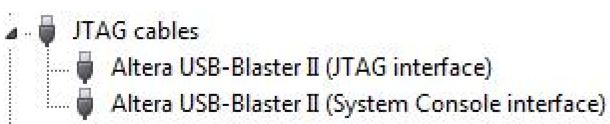
Feel free to explore the hardware source files, namely the top level file **`deca_golden_top.v`**. In this file, you'll see the top-level ports which are connected to pins on the DECA board, reg/wire declarations and some structural coding including instantiations for two components. One of these, named **`RF0002_BASE`**, contains a Nios II soft core processor that will be running the software code.

7.2.3 Download the hardware configuration file (.sof) to the MAX10

7.2.3.1 Open the Quartus II Programmer from **Tools → Programmer** or double-click on Program Device (Open Programmer) from the Tasks pane. The Programmer should open with a pre-defined configuration showing the appropriate device and programming file selected.



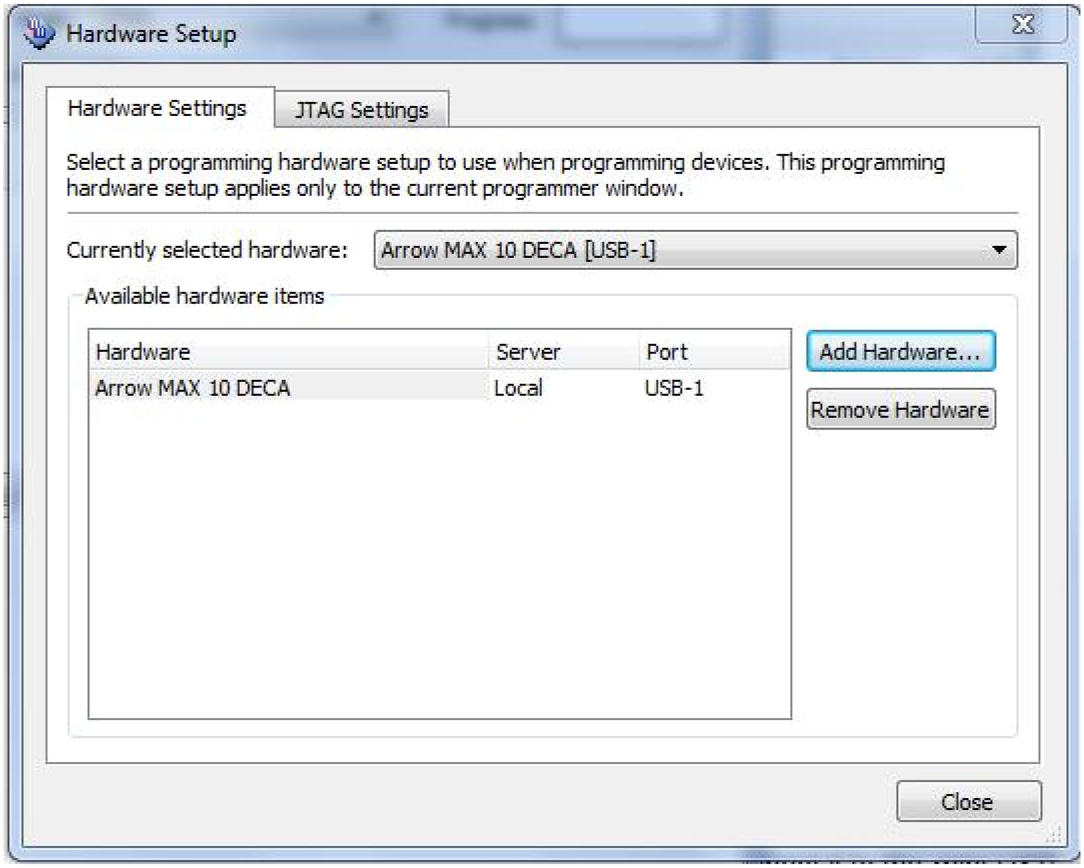
7.2.3.2 In the same orientation as above, plug in the USB cable included with your kit to the top USB connector labeled **`J10 UB2`**. Since the USB Blaster II driver software should already be installed, the Window's Device Manager should display two entries under "JTAG Cables".



You should see a few LEDs light up on your DECA including the blue LED labeled **`3.3V`** and the green LED labeled **`CONF_D`**.

If the Device Manager shows an unconfigured USB Blaster, if Windows tries to look for drivers, or if the LEDs on the DECA do not light up, ask your workshop trainer for help.

7.2.3.3 Click the Hardware Setup… button in the upper left of the Programmer window. Double-click the Arrow MAX10 DECA entry in the Hardware pane. The Currently Selected Hardware drop-down should now show Arrow MAX10 DECA [USB-1]. Depending on your PC, the USB port number may be different. Click Close.

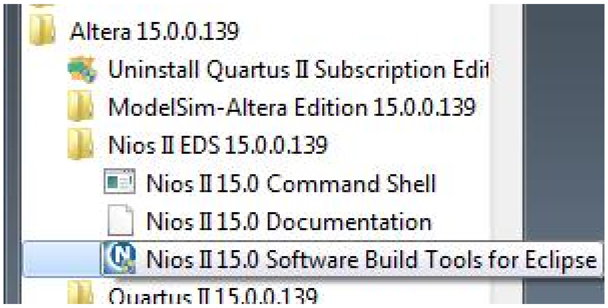


7.2.3.4 Make sure the Program/Configure checkbox is checked and click Start to program the DECA. You should see the **CONF_D** LED illuminate to indicate that the configuration is complete and the Progress bar in the Quartus II programmer should reach 100% (Successful). Additionally, all 8 blue LEDs should be on.

7.2.4 **Download the software executable (.elf) to the Nios II Soft Processor**

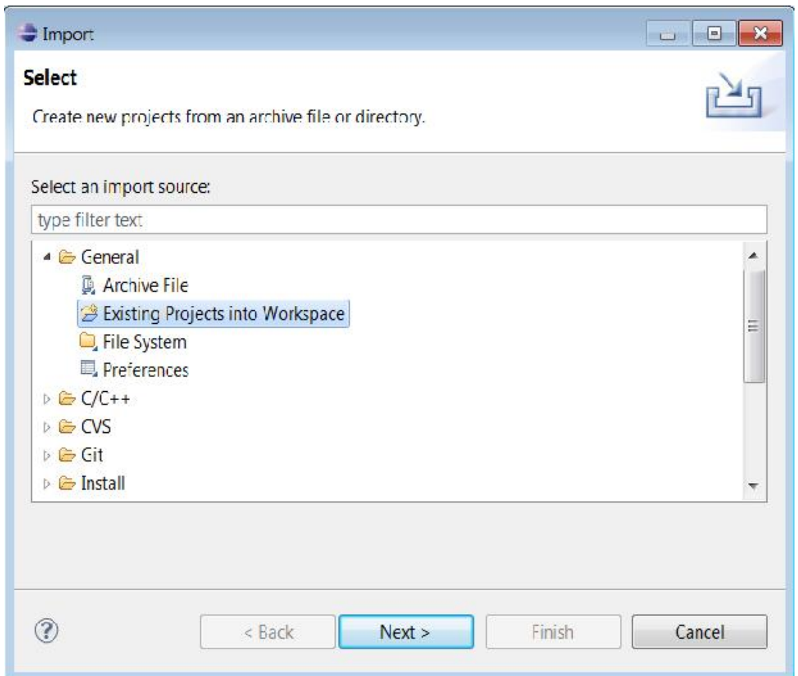
Now that the MAX10 has been programmed with the necessary hardware configuration, the software executable needs to be downloaded to the memory for the Nios II processor to begin executing it.

7.2.4.1 Launch Nios II 15.0 Software Build Tools



7.2.4.2 Import both the application and the bsp project folders using the menu: **File → Import**

7.2.4.3 The Import dialog box appears. Select **General → Existing Projects into Workspace** and Click Next

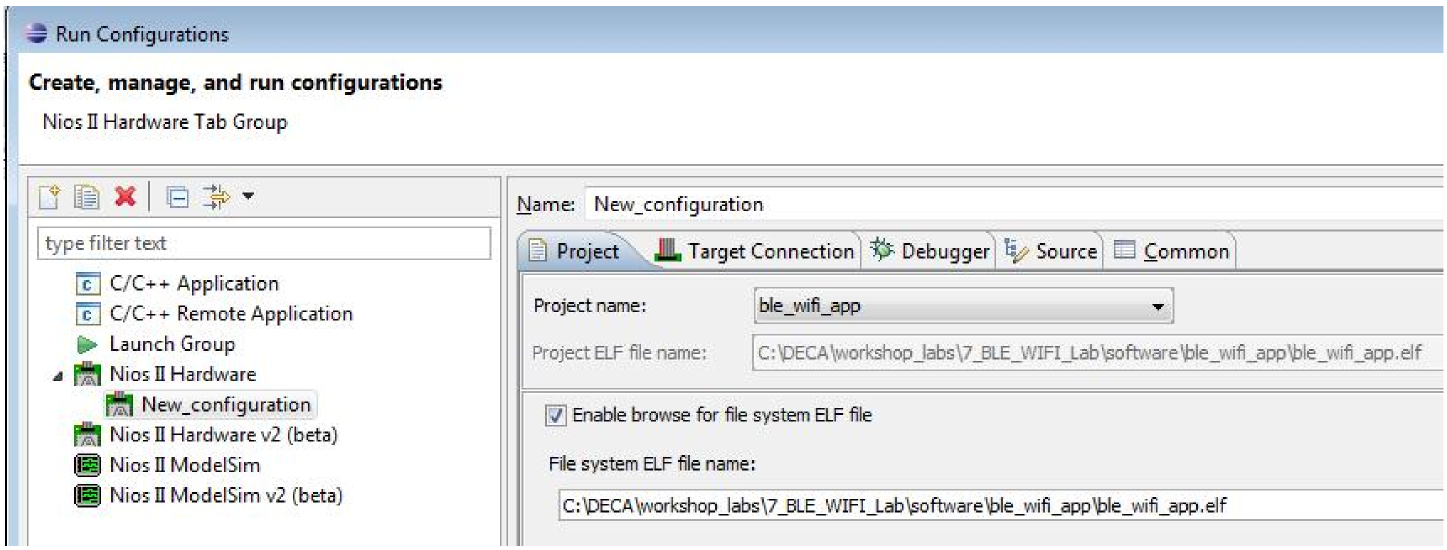


7.2.4.4 Browse to **`C:\DECA\workshop_labs\7_BLE_WIFI_Lab\software`** folder and select both **`ble_wifi_app`** and **`ble_wifi_bsp`** projects. Select "Finish".

7.2.4.5 For this step, we will not build the software project but merely just run the existing elf (executable linked format) file.

7.2.4.6 In the menu: **Run → Run Configurations…** Click **Nios II Hardware** along the left pane, and press the new button ()

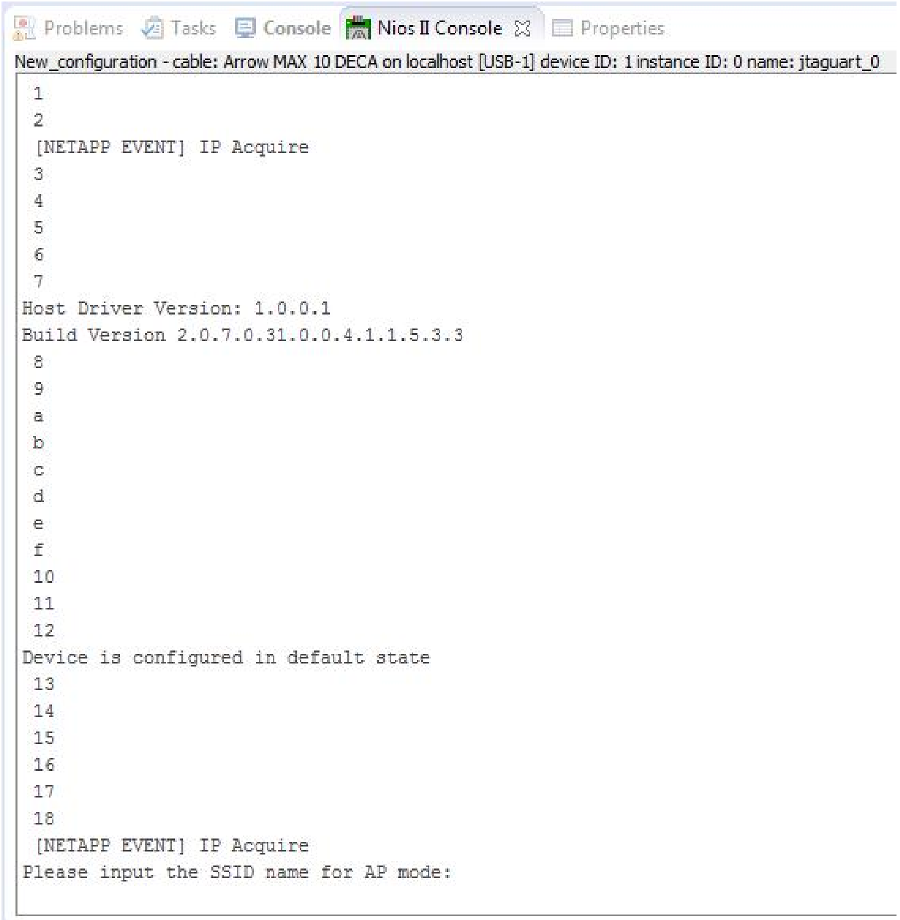
7.2.4.7 A new configuration will appear. In the Project tab on the right-hand pane, check the Enable browse for system ELF file and browse to the **`ble_wifi_app.elf`** file as shown:



7.2.4.8 In the Target Connection tab, click the Refresh Connections if the Run button has been grayed out.

7.2.4.9 Click Run to launch the ble_wifi_app.

When the ble_wifi_app launches, it will run through several lines of code before stopping and asking the user for some input as shown



7.2.4.10 Create a custom SSID for the WIFI access point. Enter any name you wish, up to 32 characters long, then press enter

7.2.4.11 Enter what type of encryption you want for this session. Press **1** for Open encryption

7.2.5 Interact with the DECA board over Wi-Fi

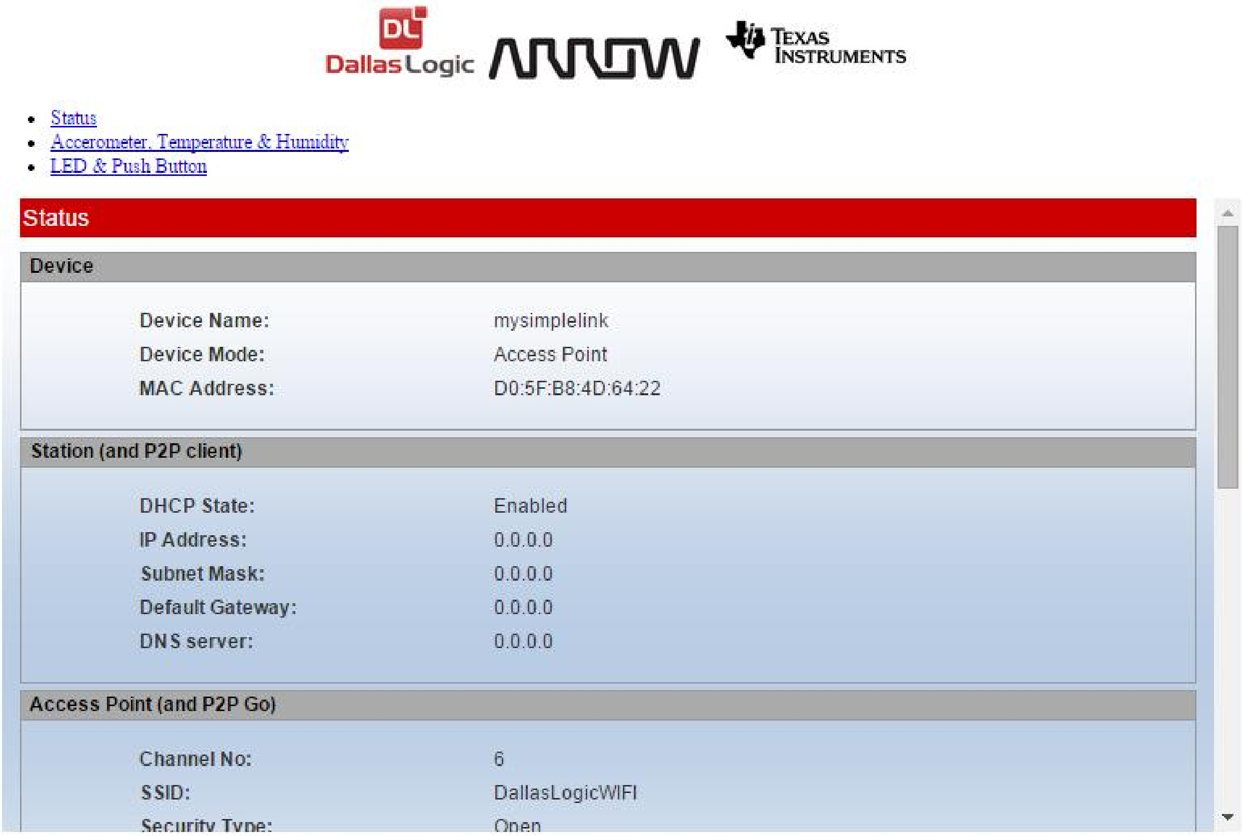
The TI CC3100 on the BLE/Wi-Fi cape has now been configured as a Wi-Fi access point (AP). In this section, you will connect to the AP and send/receive data to the DECA board over the link.

7.2.5.1 From your smartphone or laptop, connect to SSID you created just as you would with any other wireless network. The access point has no security so your device should have no trouble connecting.

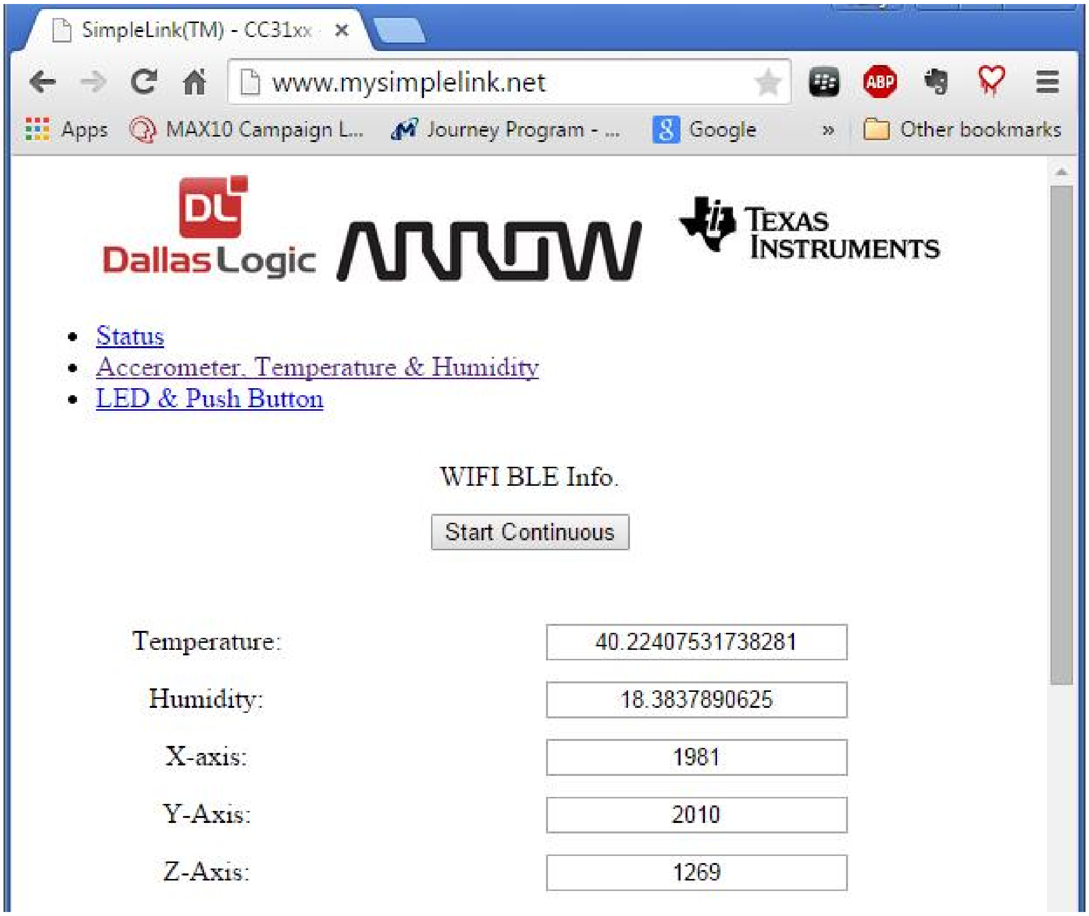
The Nios II Console should report a few more lines of information showing that the AP lease has been acquired and a client is connected.

```
 [NETAPP EVENT] Lease Aquire
 1a
 1b
Configured CC3100 in AP mode, Restarting CC3100 in AP mode
 1c
 1d
 1e
 1f
 [NETAPP EVENT] IP Acquire
Connect client to AP test

 20
 21
 [NETAPP EVENT] Lease Aquire
Client connected
 22
 23


 Domain name = mysimplelink.net 24

 Device URN = mysimplelink
```

7.2.5.2 In your web browser, type the domain name given by the terminal into the URL field i.e. **`www.mysimplelink.net`**. You should see the following web page showing status information for the Wi-Fi access point. Feel free to scroll down and explore the Status page.



7.2.5.3 Click the link for the accelerometer, temperature & humidity data. The new page displays instantaneous accelerometer, temperature and humidity data from the various sensors on the board. A live feed of this data can be enabled by clicking the "Start Continuous" button on the page.



Tilt the DECA board in various directions to see if you can determine the X, Y and Z orientations of the board.

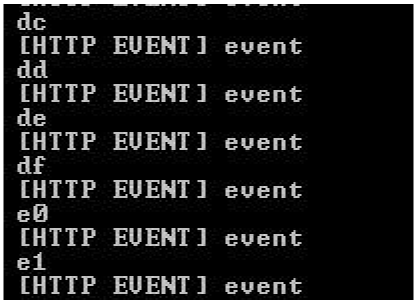
7.2.5.4 Click the link for the LED & Push Button data. The new page displays the status of the LEDs on the DECA

Note that the Push Button portion is currently not implemented.

NOTE: Pushbutton 0 [Key 0] is connected to the system reset. Pushing this button will cause Nios to reset. If you accidentally reset the system, restart the application (Run →Run Configuration)

Click a few LED buttons in the webpage and observe the corresponding LEDs illuminating on the DECA board.

Notice that the Nios II terminal displays an [HTTP EVENT] entry for each action you make in the webpage.



Congratulations! You have completed the WiFi portion of this lab!

7.2.6 **Interact with the DECA board over Bluetooth® LE**

The CC2650 fully implements the Bluetooth® Low Energy stack and is configured to appear as a pariable device to other Bluetooth capable devices. In this section, you will pair your smartphone with the DECA and interact with the sensors over the Bluetooth link.

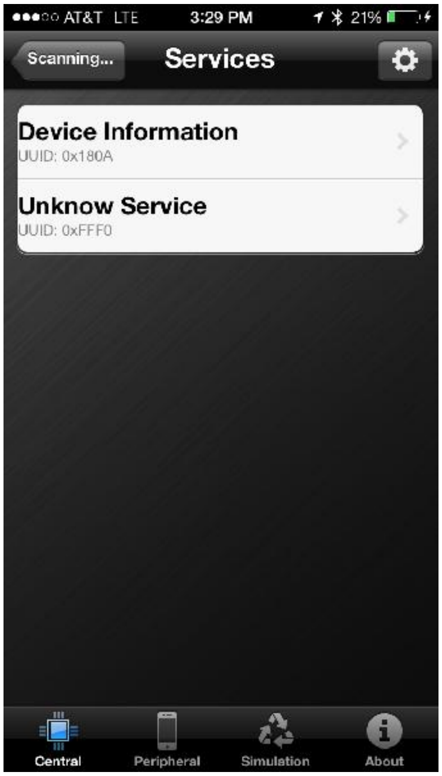
If you have an older smartphone, your device may not support Bluetooth Low Energy. Please ask your workshop trainer for help. For iOS users, continue on to the next step. For Android users, please skip to section 7.2.6.7.

7.2.6.1 Open the Bluetooth settings on your iOS device and ensure that the device's Bluetooth function is ON.

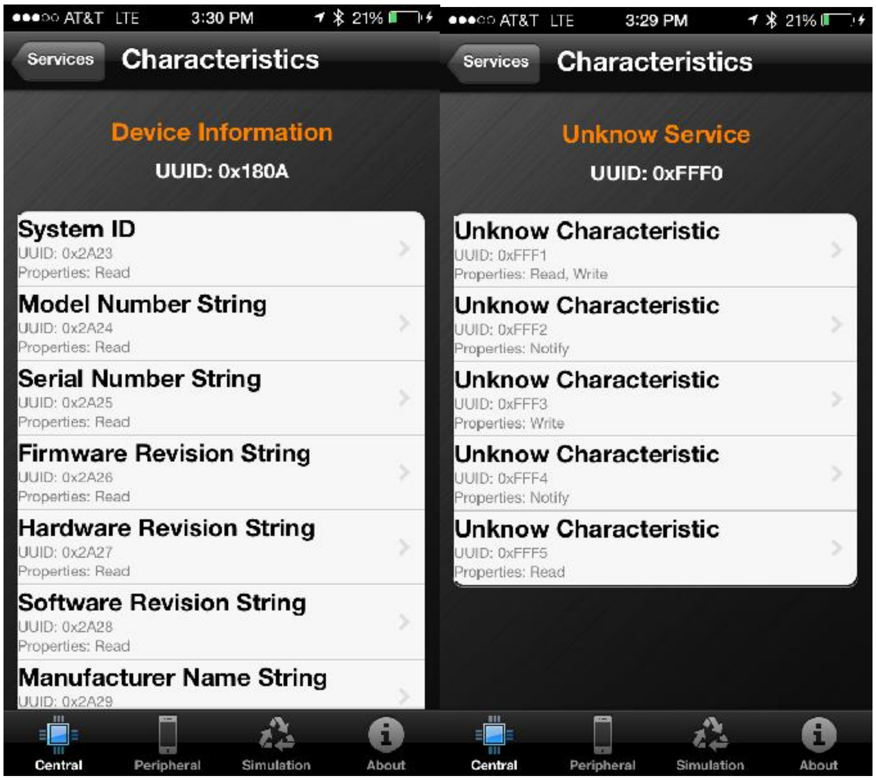
7.2.6.2 Open the App store and search for **`BLE Utility`**. Install the app below on your device.

7.2.6.3 After the installation is complete, open the app. After the application opens and scans, you should see at least one entry for ARROW BLE WIFI CAPE. Note the negative number in the blue box. This is the device's Received Signal Strength Indicator (RSSI) number. It measures the strength of the Bluetooth device's signal in dB.

7.2.6.4 If there are multiple entries, move your iOS device very close to your DECA board. You should observe the RSSI value of one of the entries approach about **`-30`**. Select this device to pair with it.



The CC2650 has been configured to provide two data "services". One provides Device Information such as system ID, hardware revision, and serial number; each of these fields is called a "characteristic". The other, titled "Unknown Service", allows the BLE Utility app to read temperature, humidity, and accelerometer data from the DECA board.



Feel free to explore the Device Characteristics service to see what kind of information can be learned.

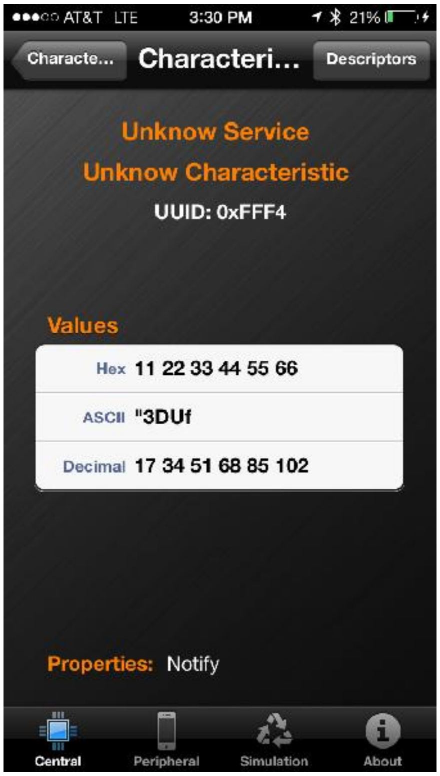
7.2.6.5 From the "Unknown Service" select the Unknown Characteristic with UUID of 0xFFF2. This characteristic displays the raw temperature and humidity data from the DECA board.

You should see the hexadecimal value fluctuating every few seconds. The format of the data is as follows:

temperature data (2 bytes), Humidity data (2 bytes), 0x0000

Wrap your hands around the DECA board and blow hot air on the board and observe how these values change.

7.2.6.6 Go back to the "Unknown Service" and select the Unknown Characteristic with UUID of 0xFFF4. This characteristic displays the raw accelerometer data from the DECA board.



Again, you should see the hexadecimal value fluctuating. The format for the accelerometer data is as follows:

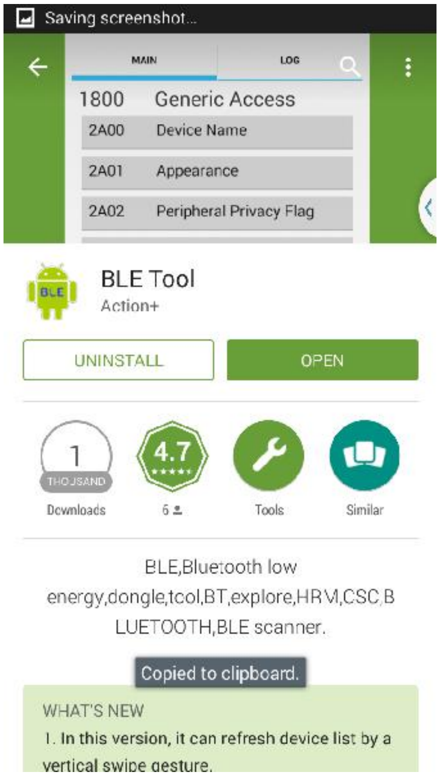
x-data (2 bytes), y-data (2 bytes), z-data (two bytes)

Tilt the DECA around and watch the hex value change. See if you can determine the nominal orientation of the accelerometer!

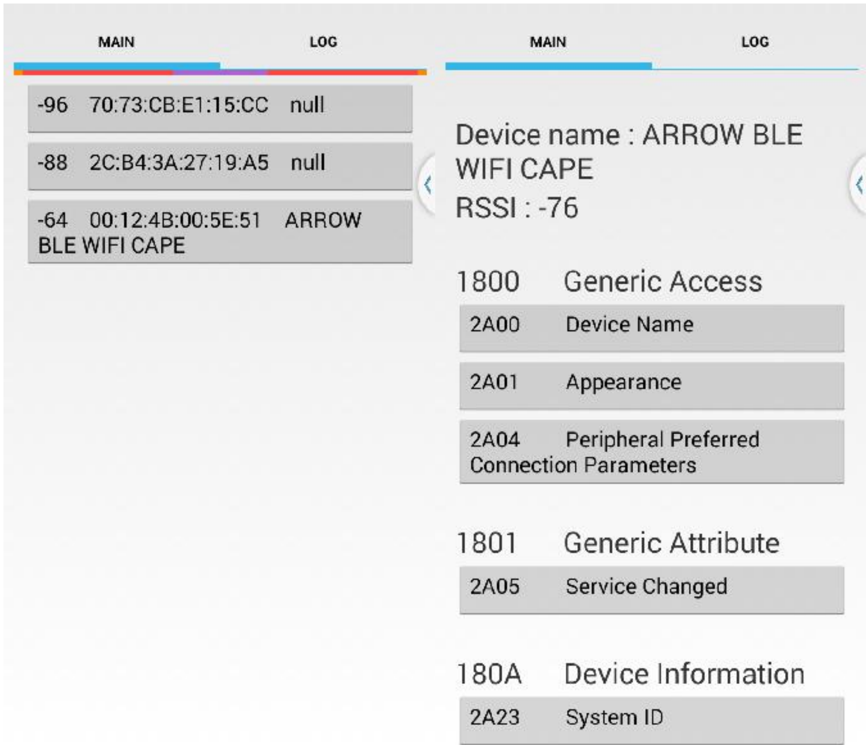
Congratulations! You have completed the BLE portion of this lab!

7.2.6.7 Open the Bluetooth settings on your Android device and ensure that the device's Bluetooth function is ON.

7.2.6.8 Open the Play Store and search for **`BLE Tool`**. Install the app below.



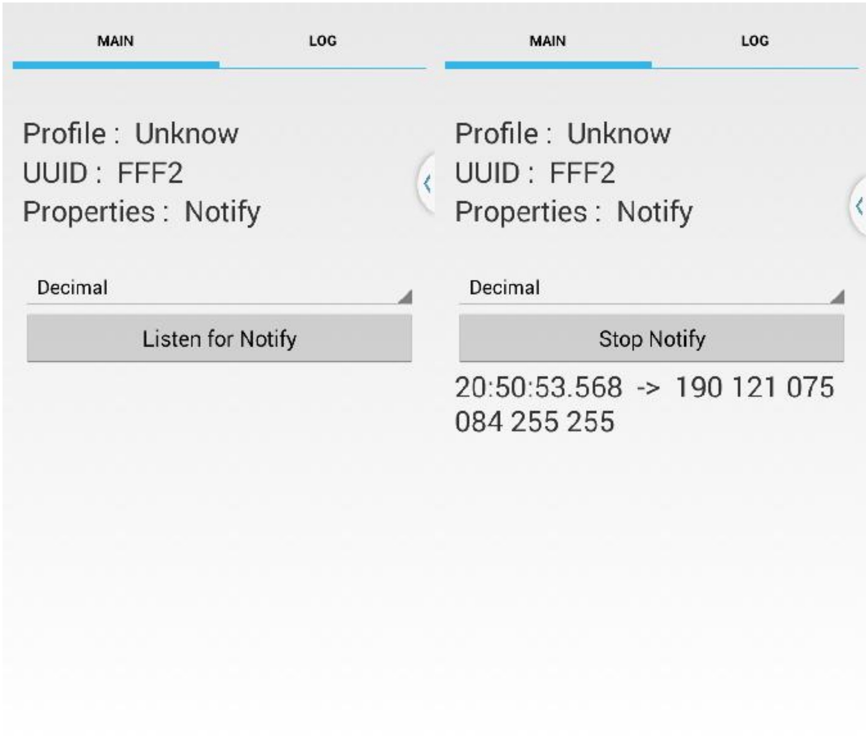
7.2.6.9 After the installation is complete, open the app. After the application opens and scans, you should see at least one entry for ARROW BLE WIFI CAPE. Note the negative number on the left side of each entry. This is the device's Received Signal Strength Indicator (RSSI) number. It measures the strength of the Bluetooth device's signal in dB.



7.2.6.10 If there are multiple entries, move your Android device very close to your DECA board. You should observe the RSSI value of one of the entries approach about **-30**. Select this device to pair with it.

7.2.6.11 The CC2650 has been configured to provide two data "services". One provides Device Information such as system ID, hardware revision, and serial number; each of these fields is called a "characteristic". Feel free to open some characteristics and browse the information available. The other, titled "Unknown Service", allows the BLE Utility app to read temperature, humidity, and accelerometer data from the DECA board.

7.2.6.12 From the "Unknown Service" section select the Unknown Characteristic with UUID of 0xFFF2. Click the Listen for Notify button. This characteristic displays the raw temperature and humidity data from the DECA board.



You should see the hexadecimal value fluctuating every few seconds. The format of the data is as follows:

Timestamp → temperature data (2 bytes), Humidity data (2 bytes), 0xFFFF

Wrap your hands around the DECA board and blow hot air on the board and observe how these values change.

7.2.6.13 Go back to the "Unknown Service" and select the Unknown Characteristic with UUID of 0xFFF4. This characteristic displays the raw accelerometer data from the DECA board.

Again, you should see the hexadecimal value fluctuating. The format for the accelerometer data is as follows:

Timestamp -> x-data (2 bytes), y-data (2 bytes), z-data (two bytes)

Tilt the DECA around and watch the hex value change. See if you can determine the nominal orientation of the accelerometer!

CONGRATULATIONS! YOU HAVE COMPLETED THE BLE/WIFI CONNECTIVITY LAB!