



iCE40 Ultra™ RGB LED Driver Using BLE User Guide

UG109 Version 1.0, June 2015

Introduction

This guide describes how to use the iCE40 Ultra™ Mobile Development Platform for demonstrating the RGB LED controller design for user application. This guide familiarizes you with the process of setting up your RGB LED controller design environment. It guides you through the hardware and software required to successfully run your RGB LED controller demonstration.

After you complete the procedures in this guide, you will be able to:

- Set up the iCE40 Ultra Mobile Development Platform properly and become familiar with its main features.
- Work and become familiar with the software required for RGB LED Driver demo.
- Understand the design details of the RGB LED Driver demo implemented on iCE40 Ultra.
- Use other Lattice documentation in conjunction with this guide.

This document assumes that you have already installed the Lattice iECube2 and the Lattice Diamond® Programmer software and are familiar with basic tasks. If more information needed on the aforementioned software, please refer to the iECube2 and Diamond Programmer help.

For details on specific board features and other information, see the [References](#) section.

This document is divided into two sections. The first section describes the RGB LED Driver design and the second section describes the RGB LED Driver demo in detail. The RGB LED Driver demo is performed using UART interface with the processor.

Demo Package Inclusions

- Verilog source code for the demo logic design
- Lattice iECube implementation project file (.prj) for the demo project
- Aldec® Active-HDL™ simulation scripts (.do) and a Verilog test bench
- iCE40 Ultra RGB LED Driver demo bit stream file (.bin)

Demo Design Hardware Requirements

- iCE40 Ultra Mobile Development Platform Board
- Windows PC or Linux machine for implementing the demo project and downloading the bit stream
- USB cable for programming the device

Demo Design Software Requirements

- Lattice iECube design software, release August 2014 or later.
- Lattice Diamond Programmer software 3.3 or higher.

Overview

The iCE40 Ultra Mobile Development Platform provides functionality to drive multi-color LED (RGB), with individual brightness control through Pulse Width Modulation (PWM), automatic blinking control and optional breath on/off control. The iCE40 Ultra device is programmed with design that provides an UART interface, data control, buffer and LED control to drive LED. The iCE40 Ultra device can be interfaced with the application processor.

Features

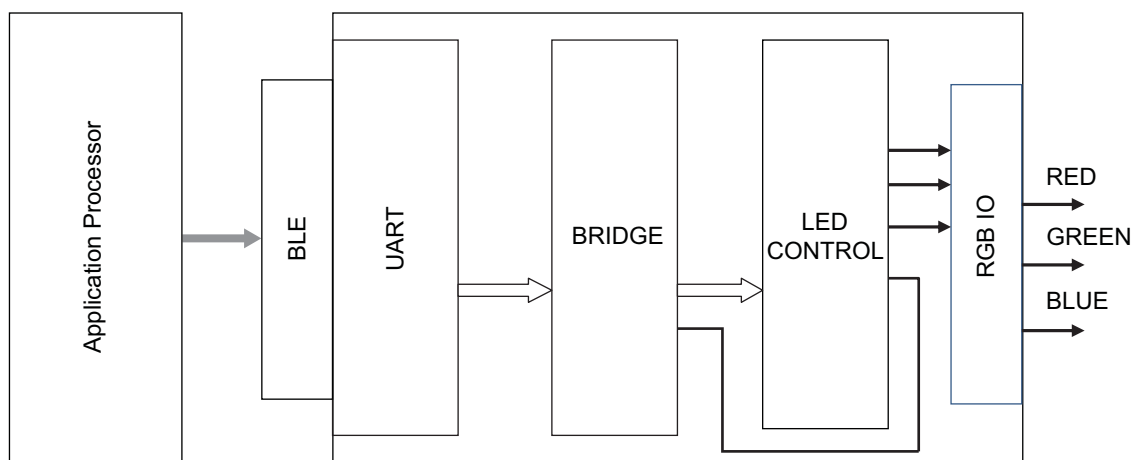
- UART interface
- LED control logic provides necessary PWM function
- RGB driver IOs with constant current sink for driving LEDs
- Control RGB LED operations through LED control logic such as
 - Individual LED brightness control for 256 levels (16M colors total)
 - On/Off timing blink rate
 - Breath Control
 - Linear and Pseudo counter mode for PWM allowing Spread Spectrum Output

Block Diagram

The Figure 1 shows the functional block diagram of the RGB LED Driver demo. The design has three main modules:

- UART Rx
- Bridge
- LED Control FSM

Figure 1. Functional Block Diagram of the RGB LED Driver Demo Design



Functional Description

This sub-section describes the function of each sub-block in inside the RGB LED Driver demo. Many of these blocks have HDL module associated with them.

Top Module

RGB LED Driver top level module contains the UART interface, the bridge module, and LED Control FSM. It also contains the RGB Driver with constant current sink. The RGB LED driver design operates after system reset has been completed.

UART Interface

The design supports UART Lite Transceiver operating at a baud rate of 4800 bps. It features a data width of 8 bits, single stop bit and parity which is disabled.

Bridge

This module reads data available on the UART bus when rx_ready goes high; the rx_data is converted to 16 bits and written into the FIFO. Whenever there is data in the FIFO, the state machine reads data from FIFO and maps it to the LEDDA IP in the following manner:

```
ledd_addr [3:0] = fifo_rd_data [3:0]
```

```
led_data [7:0] = fifo_rd_data [15:8]
```

The state machine generates data enable control signal DEN based on ledd_on output from THE LEDD IP. The state machine waits for the ledd_on to go low and then writes the buffered commands in the FIFO into the LED control registers.

LED Control

This is the LED control RTL design with 4-bit internal registers for adjusting:

- Individual RGB color
- Blinking rate
- Brightness control

All the registers can be programmed using UART/BL.

Table 1. SB_LEDDA_IP

S. No	Address	Register	Access	Size (Bits)	Description
1	0x08	LEDDCR0	W	8	LED Driver Control Register 0
2	0x09	LEDDBR	W	8	LED Driver Pre Scale Register
3	0x0A	LEDDONR	W	8	LED Driver ON Time Register
4	0x0B	LEDDOFR	W	8	LED Driver OFF Time Register
5	0x05	LEDDBCRR	W	8	LED Driver Breath On Control Register
6	0x06	LEDDBCFR	W	8	LED Driver Breath Off Control Register
7	0x01	LEDDPWRR	W	8	LED Driver Pulse Width Register for PWMOUT0
8	0x02	LEDDPWRG	W	8	LED Driver Pulse Width Register for PWMOUT1
9	0x03	LEDDPWRB	W	8	LED Driver Pulse Width Register for PWMOUT2

SB_RGBA_DRV

This primitive instance for the IO block that provides a constant sink current open drain driver. For signal description please refer SB_RGBA_DRV usage model.

Configurable Parameters:

```
defparam RGB_DRIVER.RGB0_CURRENT = "0b111111";
```

```
defparam RGB_DRIVER.RGB1_CURRENT = "0b111111";
```

```
defparam RGB_DRIVER.RGB2_CURRENT = "0b111111";
```

These parameters control the number of current sinks providing a constant sink current from 4 mA to 24 mA in steps of 4 mA.

Operating Sequence

1. The App Proc. writes 2 bytes of data with bit [3:0] of first byte containing the LEDDA register address and the LED control register address and the next byte containing the data being written to that register.
2. The bridge reads data coming from the UART interface, converts it to 16 bit and writes it into the FIFO
3. When FIFO empty goes low the bridge drives ledd_exe low and waits for the ledd_on to go low.
4. When ledd_on becomes low the bridge state machine writes the commands buffered in the FIFO in the LEDDA registers.
5. When FIFO is empty and all data has been written to the led registers the state machine drives ledd_exe high and LED control continues with execution of the PWM sequence.

Signal Description

Table 2 lists the external interface signals.

Table 2. External Interface Signals

Signal Name	Pin Type	Pull-Up Required	CM36A Pin Assignment	Signal Description
clki	IN	Yes	[F4]	On board clk
LED Interface				
red	OUT	Yes	[C6]	PWM output from RGB driver to red of RGB LED
green	OUT	Yes	[B6]	PWM output from RGB driver to green of RGB LED
blue	OUT	Yes	[A5]	PWM output from RGB driver to blue of RGB LED
Application Processor Interface				
i_rx	IN	Yes	[E3]	Input to UART

Resource Utilization

Table 3. Resource Utilization

Device	LUTs	Registers	Memory	I/Os	LEDDA	RGBA_DRVs
iCE5LP4KSWG36	373	212	0	2	1	1

Running the Simulation

To run the simulation:

1. Open Active HDL.
2. Copy the full path for the folder containing the run.tcl file (hardware/simulation).
3. Go to the location through Active HDL console (cd full path).
4. Go to **Tools > Execute macro**.
5. Select the run.tcl script option.

The simulation runs automatically.

RGB LED Controller Demonstration

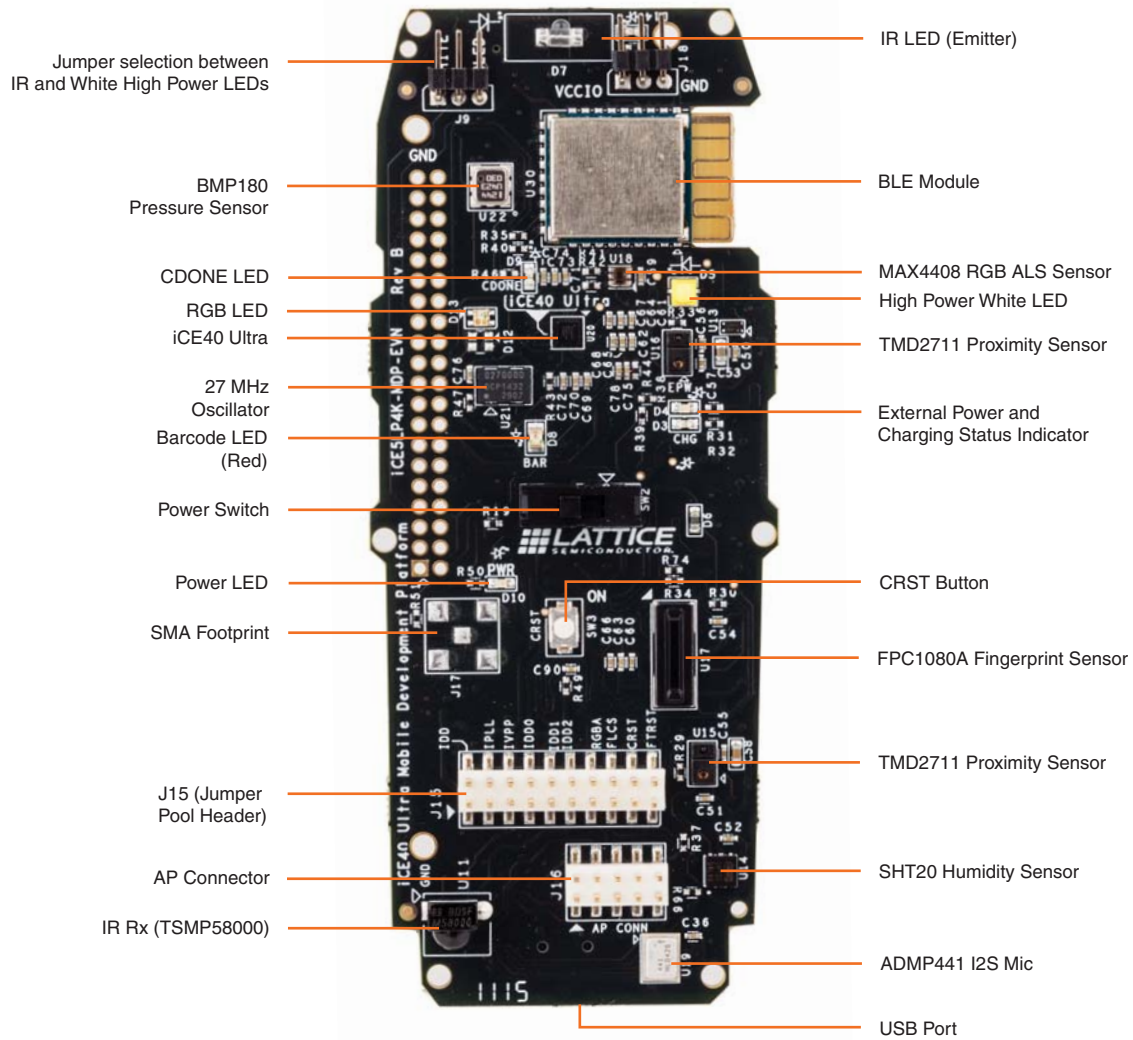
Figure 2. RGB LED Controller Demonstration



iCE40 Ultra Mobile Development Platform Details

The details of the iCE40 Ultra Mobile Development Platform are shown in Figure 3.

Figure 3. iCE40 Ultra Mobile Development Platform Details



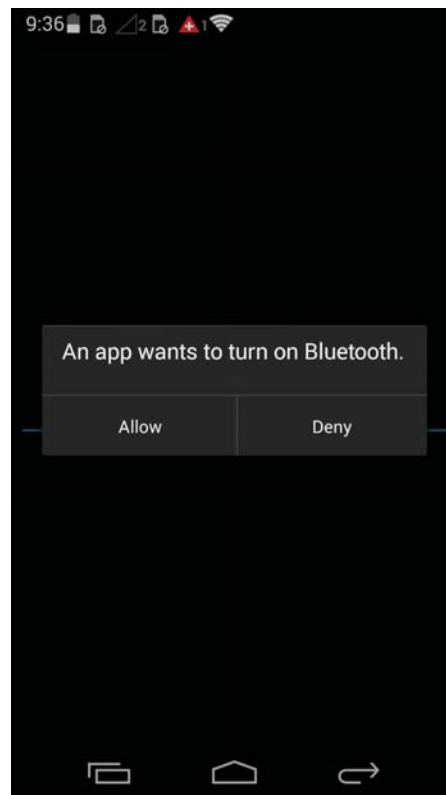
Connecting the iCE40 Ultra Mobile Development Board to the Android Phone

To connect the iCE40 Ultra Mobile Development Platform to the Android phone:

1. Turn on the mobile development board using switch SW2.
2. If Bluetooth is already enabled on the mobile, go to Settings and disable it.
3. Open the installed application by going to the Apps menu and clicking **RGB LED Controller BLE demo**.

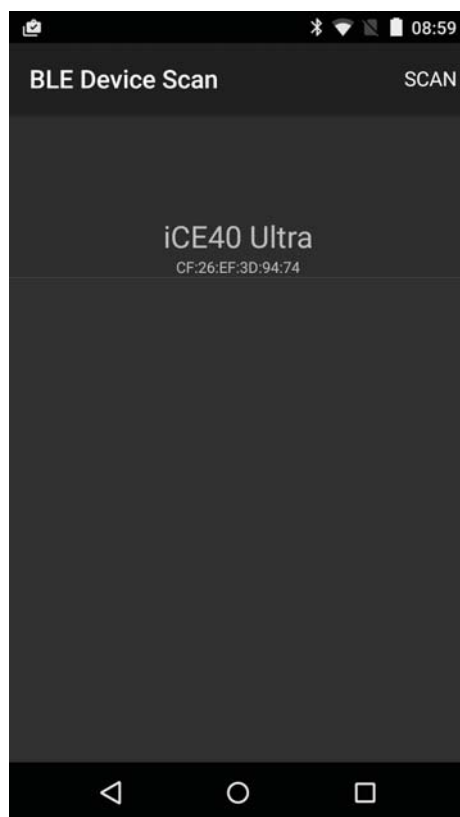
A pop-up window appears asking for permission to activate Bluetooth on the phone.

Figure 5. Activating Bluetooth



4. Click **Allow** to activate Bluetooth.
5. A window appears that scans and lists BLE devices in the vicinity. Click the desired device to connect it to the phone.

Figure 6. BLE Device Scan



Note: If the BLE does not detect the iCE40 Ultra device, the BLE firmware needs to be updated. To update the BLE firmware, refer to the [Appendix A. Steps to Program the BLE Module Over-The-Air](#) section.

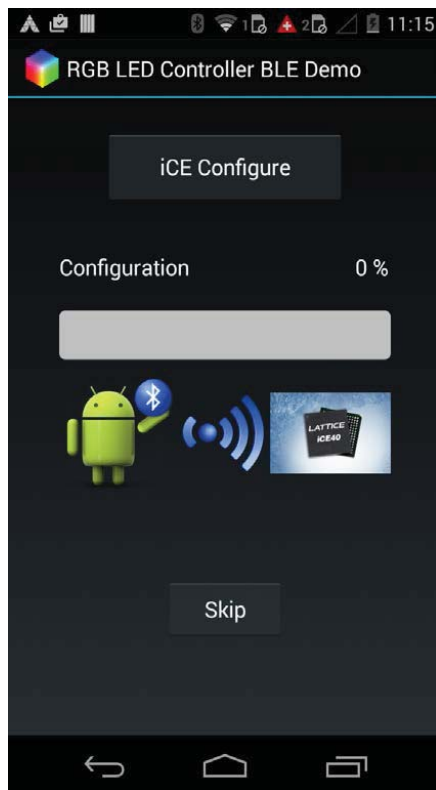
Configuring the CRAM Over-The-Air On the Android Phone

To configure the CRAM over-the-air on the Android phone:

Note: The bitstream is integrated in the .apk itself.

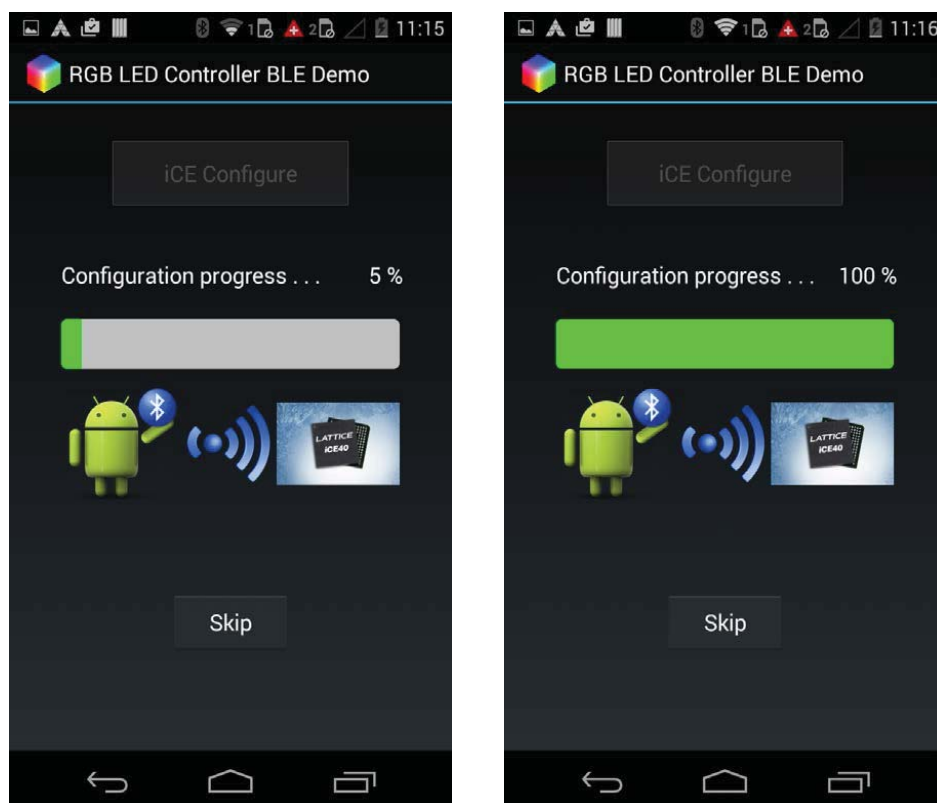
1. Once you connect to the iCE40 Ultra device, the following screen is displayed.

Figure 7. CRAM Configuration



2. Press **iCE Configure**. The configuration progress is displayed.

Figure 8. Configuration Progress



When configuration is completed, the main application screen is displayed as shown in Figure 9.

Configuration Results

The average CRAM configuration time is 35 seconds.

Demo Procedure

To run the demo:

1. When CDONE LED glows on the iCE40 Ultra board, unlock the screen if the phone is locked or else go to the Android Application menu and click the **RGB LED Controller Demo** application.
2. The application is now ready. Click on various colors to change the glowing color of the RGB LED. Other features such as brightness, blinking and breathing can be changed using the sliders. The Color Selector allows you to choose from 13 different colors in the pallet.
3. Click on the various colors to change the glowing color of the RGB LED. Other features such as brightness, blinking and breathing can be changed using the sliders.

Figure 9. RGB LED Controller BLE Demo Application



RGB LED Controller BLE Demo Application Features

- The Color Selector allows you to choose from 13 different colors in the pallet.
- The Brightness control slider allows you to control the brightness of the LEDs. Adjusting the slider to the left decreases brightness and adjusting the slider to the right increases brightness.
- The Blink rate control slider allows you to adjust the blinking interval of the LEDs. Dragging the slider farthest to the left places the blink rate value at 0, this results in no blinking. Adjusting the slider to the right increases the blink ON and OFF interval from 0.256 second to 3.84 seconds. The rate may be increased 15 times in increments of 0.256 seconds.
- The Breathing control slider allows you to adjust the breathing ramp of the LEDs. Adjusting the slider to the left decreases the speed of breathing and adjusting the slider to the right increases the speed of breathing.
Note: This step prevents burning of the IR LED due to high current.

Troubleshooting

When the mobile phone is connected to the host system, USB debugging is allowed as shown in Figure 10.

Figure 10. USB Debugging



If the Android application hangs or does not respond, perform the following steps:

1. Close the RGB LED Controller BLE Demo apk process running in background.
2. In the Android menu, select **System settings > Applications > Manage Applications > RGB LED Controller Demo > Force Stop**.
3. Open the RGB LED Controller Demo application from the Android menu. The application is ready to control the RGB on the iCE40 Ultra Lite board.

References

- DS1050, [iCE40 Ultra Family Data Sheet](#)
- iCE40 Ultra Board Schematics

Technical Support Assistance

Submit a technical support case via www.latticesemi.com/techsupport.

Revision History

Date	Version	Change Summary
June 2015	1.0	Initial release.

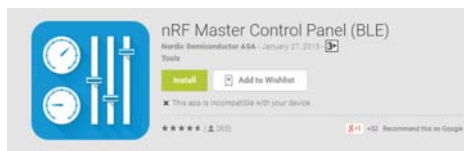
Appendix A. Steps to Program the BLE Module Over-The-Air

The iCE40 Ultra is pre-flashed Nordic S110 Soft Device 6.0, an OTA boot loader and an nRF UART application.

On the Android Phone

1. Copy the BLE firmware files (BLE_Firmware.hex and BLE_Firmware_SD7DFU.hex) into the Android phone.
2. Install [nRF Master Control Panel \(BLE\)](#) from the Android Play Store.

Figure 11. nRF Master Control Panel



On the iCE40 Ultra

To enter OTA mode:

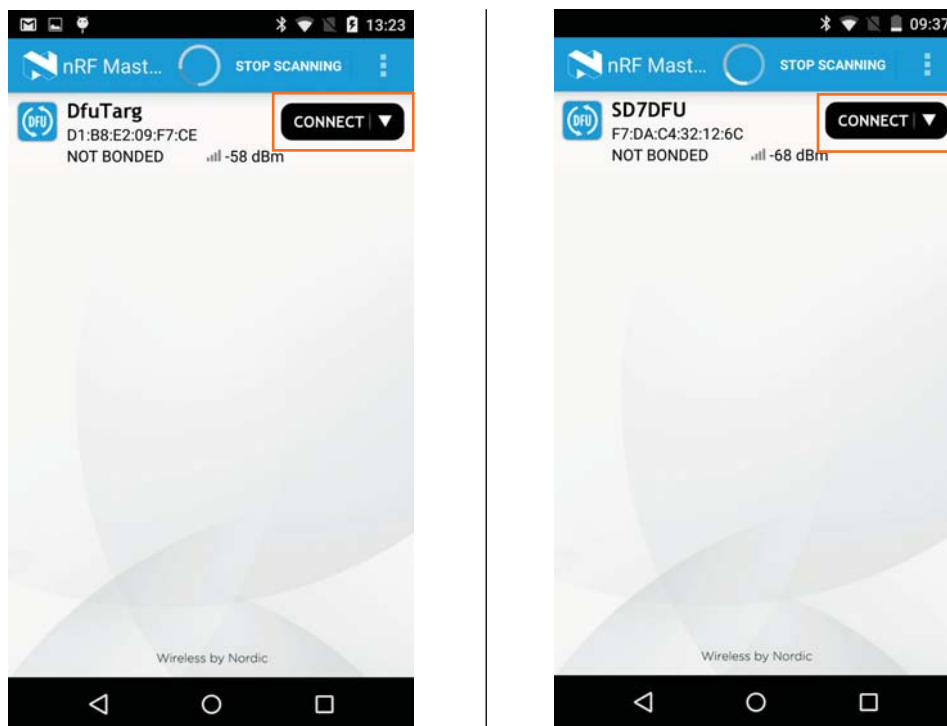
1. Join pins 2-3 on J18 (that is, connect P18 of the BLE Module to GND).
2. Power on the iCE40 Ultra board.
3. Open the installed **nRF Master Control Panel** application.

As shown in Figure 12, if the name of the BLE device is *DfuTarg*, it means that the BLE Module has Nordic S110 SoftDevice rev 6.0 installed. If the name of the BLE device is *SD7DFU*, it means that the BLE Module has Nordic S110 SoftDevice rev 7.0 installed.

Note:

The device sometimes exits the OTA mode, as it may have an internal timeout condition, causing the DFU option to become unavailable. If this occurs, retry by repowering the board.

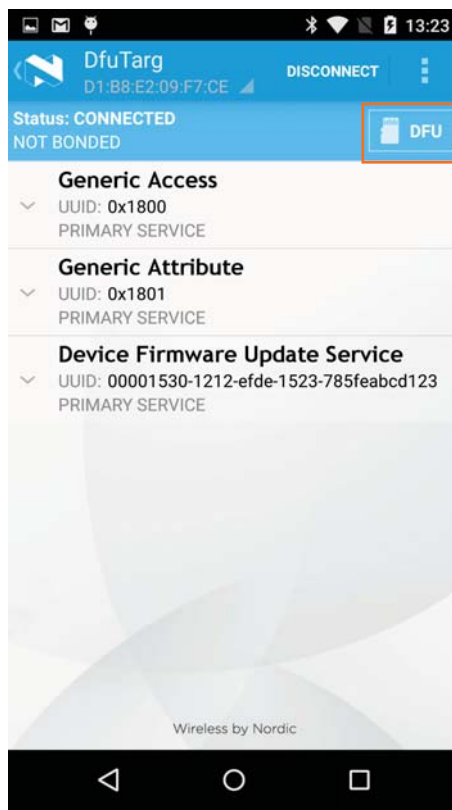
Figure 12. .DfuTarg/SD7DFU BLE Device BLE Device



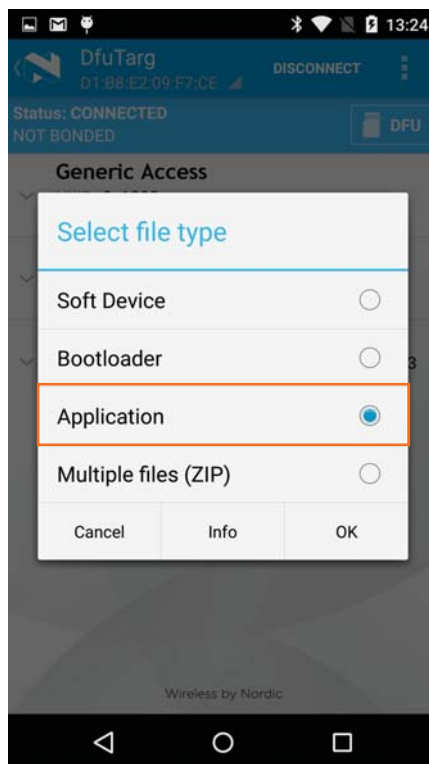
4. Press **CONNECT** to open the window shown in Figure 13.

Note: Once connected, the device may not show the DFU option at first. To show the DFU option, reconnect to the device.

Figure 13. DFU Option



5. Press the **DFU** option.
6. The resulting window, after connection is established, is shown in Figure 14. Select the file type **Application**.
7. Press **OK**.

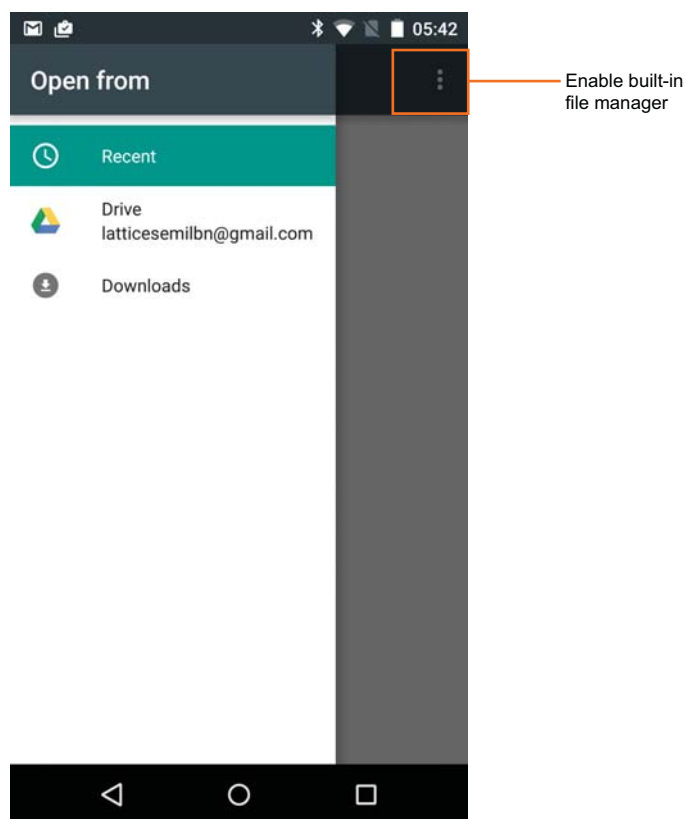
Figure 14. File Type Selection

8. Set up the file manager.

If a file manager is already installed, proceed to Step 9.

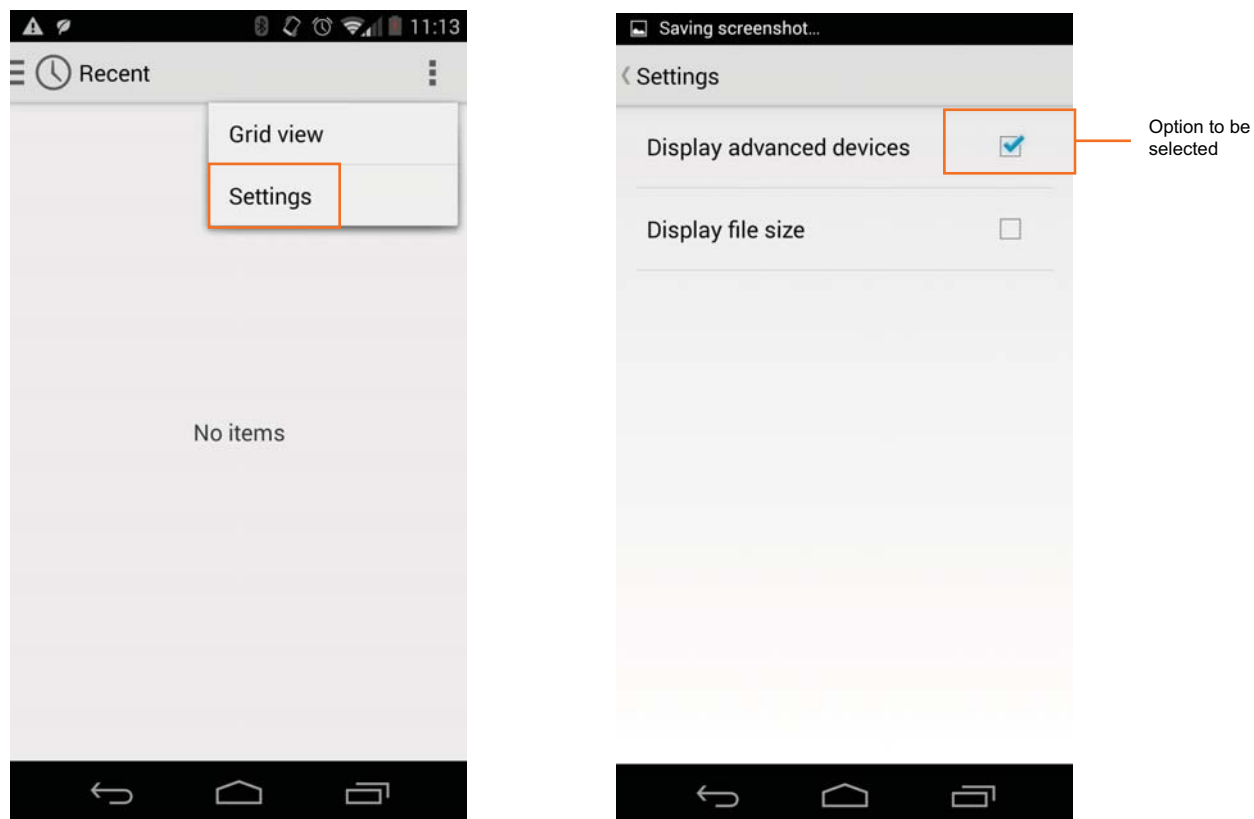
If a file manager is not yet installed on the phone, enable the built-in file manager. Press the top right corner button as shown in Figure 15.

Figure 15. File Manager Selection Window



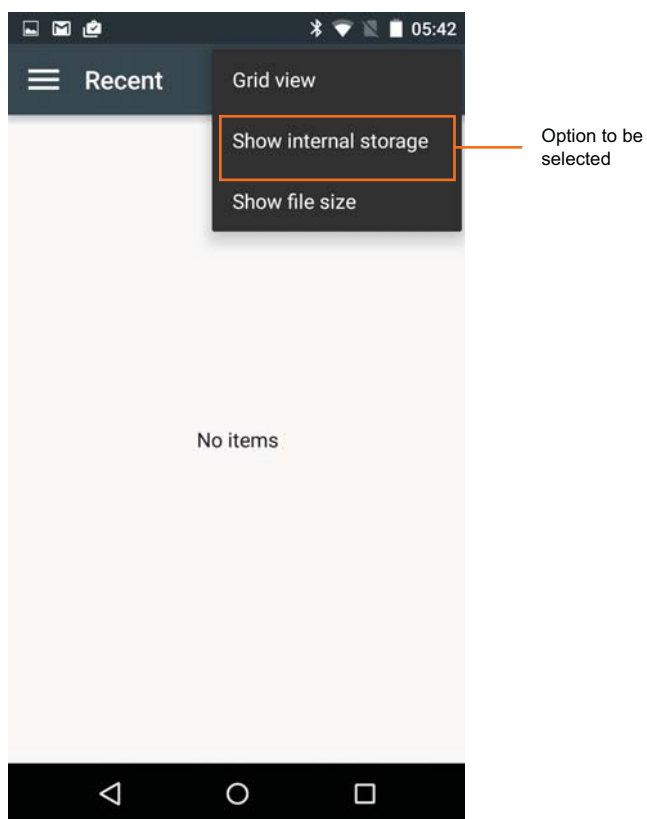
In a KitKat operating system, after pressing the top-right settings button, the resulting window is shown in Figure 16.

Figure 16. Option for KitKat 4.4 OS



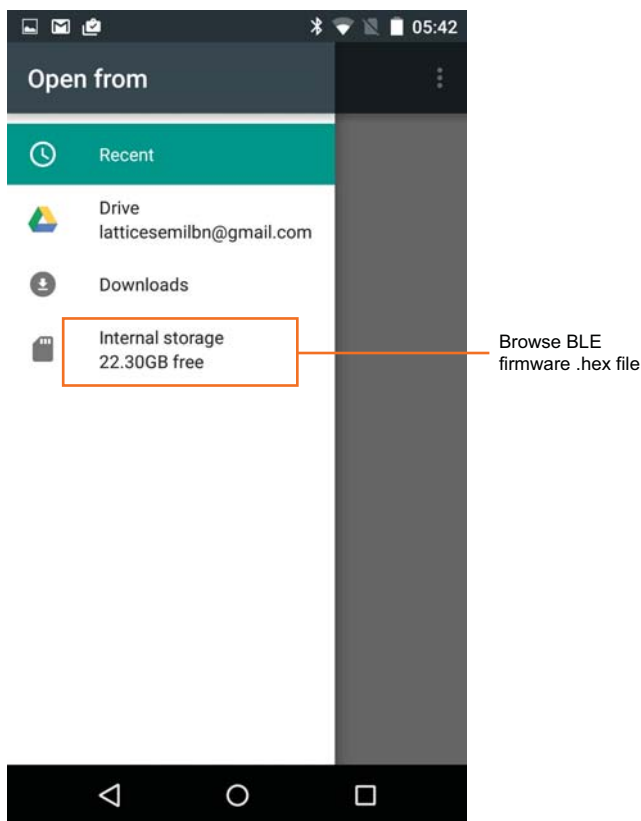
In a Lollipop operating system, the resulting window is shown in Figure 17.

Figure 17. Option for Lollipop OS



After following the above procedure, the resulting window is shown in the Figure 18.

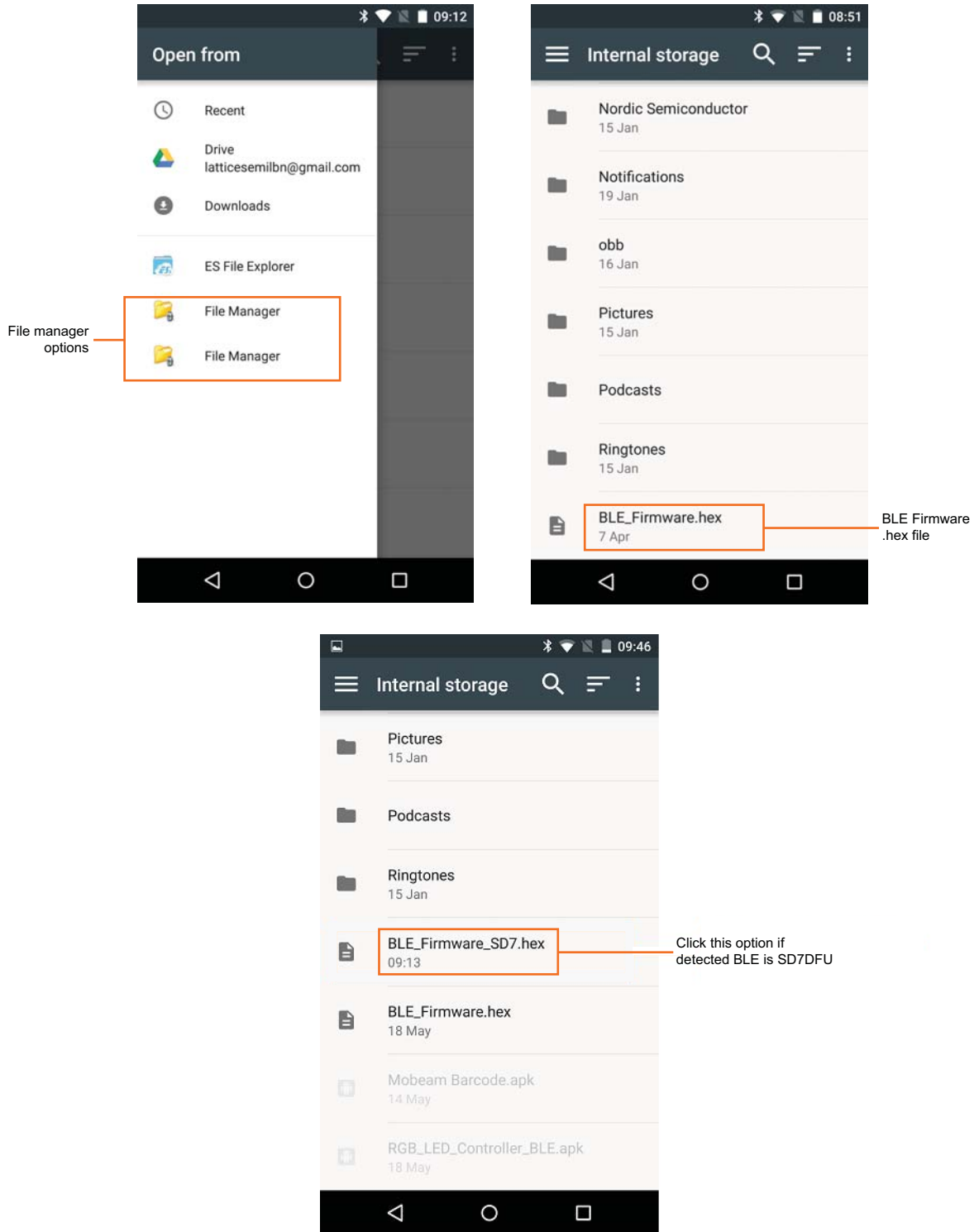
Figure 18. Enabled Built-in File Manager



9. Select the copied BLE Firmware (.hex file).

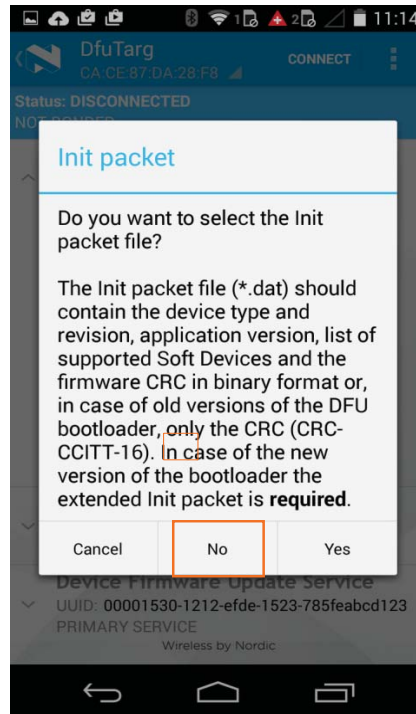
As shown in Figure 19, choose **BLE_Firmware.hex** if the device scanned is *DfuTarg* (Nordic S110 SoftDevice rev 6.0). Choose **BLE_Firmware_SD7DFU.hex** if the device scanned is *SD7DFU* (Nordic S110 SoftDevice rev 7.0).

Figure 19. BLE Firmware .hex File



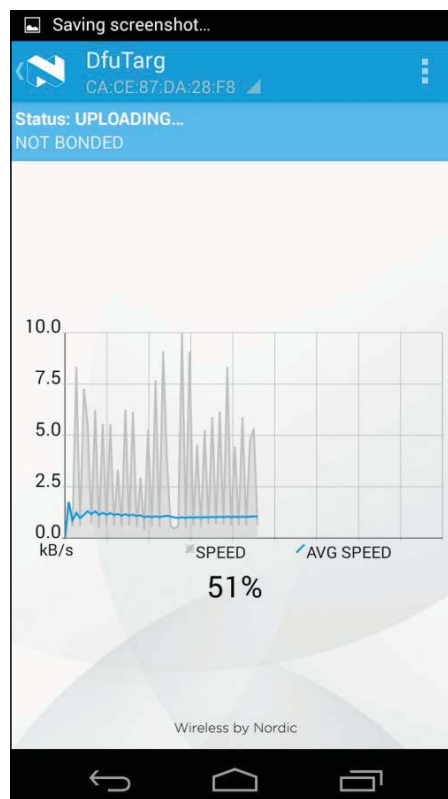
10. Select **No** as shown in Figure 20.

Figure 20. Init Packet



11. The upload progress is displayed on the interface.

Figure 21. Upload Progress



12. Join pins 1-2 on J18, and repower the board.
13. Open the installed Demo Application. The device is now scanned as iCE40 Ultra.