



iCE40 Ultra™ Self-Learning IR Remote with BLE User Guide

UG104 Version 1.0, June 2015

Introduction

This document provides technical information on iCE40 Ultra™ as a self-learning IR remote controller. This guide familiarizes you with the process of setting up your self-learning IR remote controller design environment. It guides you through the hardware and software required to successfully run your self-learning IR remote 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 self-learning IR remote controller demonstrations.
- Understand the design details of the self-learning IR remote controller demo implemented on iCE40 Ultra.
- Use other Lattice documentation in conjunction with this guide.

This document assumes that you have already installed the Lattice iCEcube2 and the Lattice Diamond® Programmer software and are familiar with basic tasks. If you need more information on these software, please refer to the iCEcube2 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 self-learning IR remote controller design and the second section describes the self-learning IR remote controller demonstration in detail. The self-learning IR remote controller demonstration is performed using a UART interface with the Application Processor.

Demo Package Inclusions

- Verilog source code for the demo logic design
- Lattice iCEcube implementation project file (.prj) for the demo project
- Aldec® Active-HDL™ simulation scripts (.do) and a Verilog test bench
- iCE40 Ultra self-learning IR remote Controller bit stream file (.bin)

Demo Design Hardware Requirements

- 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 iCEcube design software, release August 2014 or later
- Lattice Diamond Programmer 3.3

Overview

This design example enables the capabilities of the iCE40 Ultra SWG36 device as a self-learning IR remote controller.

Features

- Supports UART interface
- Configurable carrier frequency
- Self-learning capability
- Interface 16 bit address, 8 bit data

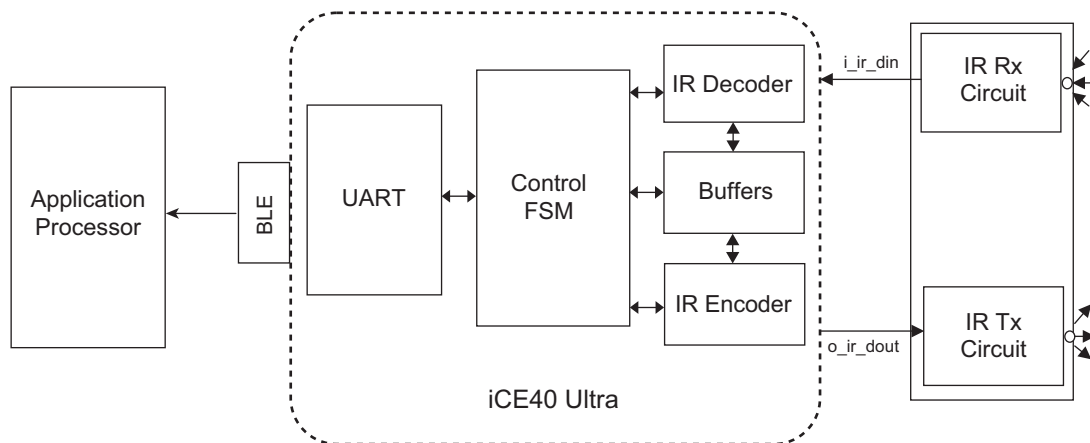
Block Diagram

Figure 1 shows the functional block diagram of the self-learning remote controller. The design has the following four main modules:

- UART interface
- Controller FSM
- IRTCV TX and RX Control FSM
- IR data buffers.

Figure 1 shows the block diagram of database based self-learning remote controller with BLE processor interface.

Figure 1. Functional Block Diagram of the IR Self-Learning Remote Controller



Functional Description

The Top Level (irda_top)

The irda_top module instantiates Serial interface module, IR buffers and Hard IRTCV TX and RX Control FSM Blocks.

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.

IRTCV TX and RX Control FSM

The IR Transceiver control FSM provides logic function to transmit and receive data through Infrared LED data link. In transmit mode, this module takes the data from Tx buffer to transmit with user specified frequency. In learning mode, this module receives data from Infrared receiver and sends the received data to Rx buffer.

Buffer Description

The Tx and Rx buffers are both 2048 bytes in size and share the same address, which is from 0x0800 to 0xff. The Tx/Rx buffer includes IR commands to describe an IR frame.

Note: MSB of IR Command represents High/Low bit. For example, the IR Command Values are 8003b(high) and 0032 (low).

Table 1. Buffer Description

Address	Name	R/W	Description
0x0800	IR Command 1 High Byte	R/W	The first command means IR LED on time, the unit is 2 μ s
0x0801	IR Command 1 Low Byte	R/W	
0x0802	IR Command 2 High Byte	R/W	The Second Command means IR LED off time, the unit is 2 μ s
0x0803	IR Command 2 Low Byte	R/W	
—	—	R/W	—
—	—	R/W	
0x0FFE	IR Command 4096 High Byte	R/W	IR LED off time the unit is 2 μ s
0x0FFF	IR Command 4096 Low Byte	R/W	

Transmit IR Frame Steps

1. Write IR Commands to Tx Buffer.
2. Write Tx Length Registers.
3. Write Carrier Divider Registers.
4. Write Command Register Bit0 to Low.
5. Write Command Register Bit0 to High.

Self-Learning Steps

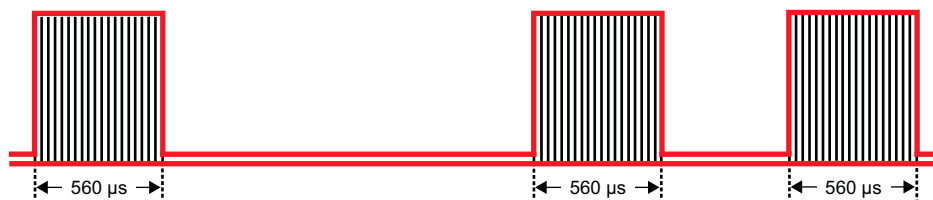
1. Write Command Register Bit1 to Low.
2. Write Command Register Bit1 to High
3. Check Status Register Bit0. High means an IR frame is stored to Rx Buffer.
4. Read Rx Length Registers to get valid bytes in Rx Buffer.
5. Read Rx Carrier Divider to get Carrier Frequency.
6. Read Rx Buffer to get a IR frame data

IR Frame Capture

Carrier Filter

To implement waveform copy process, the carrier is not required. iCE40 auto filters the carriers from i_ir_din pin. For example, get the red waveform from the original waveform. And for transmission, iCE40 auto pulls the carriers.

Figure 2. Carrier Filter



Preamble Code Detection

Different IR remote protocol includes different preamble code, so for self-learning mode, if High Level contains more than four pulses means a frame start. And preamble code also needs to be recorded.

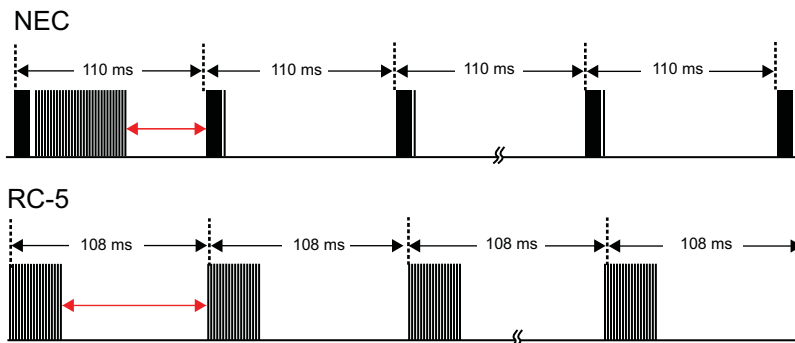
Figure 3. Preamble Code Detection



Stop Code Detection

A very long low level means a stop code. The whole frame is recorded. Overflow time is 100 ms.

Figure 4. Stop Code Detection



Clock and Reset

A system clock of 10-30 MHz is required to generate accurate carrier frequencies for remote controller. Active low reset input is required to reset the design

Table 2. Register Description

ADDRESS	NAME	ACCESS	Description
0x0001	IR Transceiver Control Register	W	Bit 7 : 0 IR Transceiver disabled 1 IR Transceiver enabled
0x0002	IR Transceiver System Clock Frequency Register 3	W	The IR systems clock frequency registers, total 28 bits, holding the binary number representing the system clock frequency in Hertz. For normal application, user should set the IRSYSFRs prior of setting the IRTCVFRs. When writing IRSYSFR3, the MSB four bits from the data bus are don't care.
0x0003	IR Transceiver System Clock Frequency Register 2	W	
0x0004	IR Transceiver System Clock Frequency Register 1	W	
0x0005	IR Transceiver System Clock Frequency Register 0	W	
0x0006	IR Tx Carrier Frequency Register 2	W	The IR Transceiver Clock Frequency Register, total 24 bits, holding the binary number representing the IR Transceiver clock frequency in Hertz. Internal clock count to generate the IR Transceiver Clock will be computed using formula as: $N = F_SYS / F_TCV$
0x0007	IR Tx Carrier Frequency Register 1	W	
0x0008	IR Tx Carrier Frequency Register 0	W	
0x000C	Tx Length Low Byte	W	Valid data in Tx buffer.(1 data = 2 Bytes)
0x000D	Tx Length High Byte	W	
0x000E	Command Register	W	Bit 0: 0 starts transmit. 1 starts self-learning.
0x1801	Rx Length High Byte	R	Valid data bytes in Rx buffer. (1 data = 2 Bytes)
0x1802	Rx Length Low Byte	R	
0x1803	Status Register	R	Bit 0: High means self-learning data have been stored to Rx Buffer.
0x0006	IR Rx Carrier Frequency Register 2	R	Carrier frequency while learning
0x0007	IR Rx Carrier Frequency Register 1	R	
0x0008	IR Rx Carrier Frequency Register 0	R	

Signal Description

Table 3 lists the external interface signals,

Table 3. External Interface Signals

Signal Name	Pin Type	Pull-Up Required	Pin Assignment	Signal Description
i_sys_clk	IN	X	[F4]	Clock input to the design, 27 MHz
o_clk_stdby	OUT	X	[D2]	Oscillator clock standby input
IR Interface				
i_irda	IN	X	[F5]	IR signal from IR receiver module
o_irda	OUT	X	[A2]	IR signal out from IR transmitter module
UART/BLE Interface				
o_tx	OUT	X	[F3]	UART Output
i_tx	IN	X	[E3]	UART Input

Resource Utilization

Table 4. Resource Utilization

Device	LUTs	Registers	Memory	IR400_DRV	IOs
iCE5LP4K	1060	545	8	1	

Running the Simulation

To run the simulation:

1. Open Active HDL.
2. Copy the full path for the folder containing the run.tcl, for example hardware/simulation.
3. Go to the location through the Active HDL console, for example the cd full path.
4. Go to **Tools > Execute macro** and select the run.tcl script.
5. The simulation runs automatically.

IR Self-Learning Demonstration

This section describes the IR Self-Learning demonstration in detail.

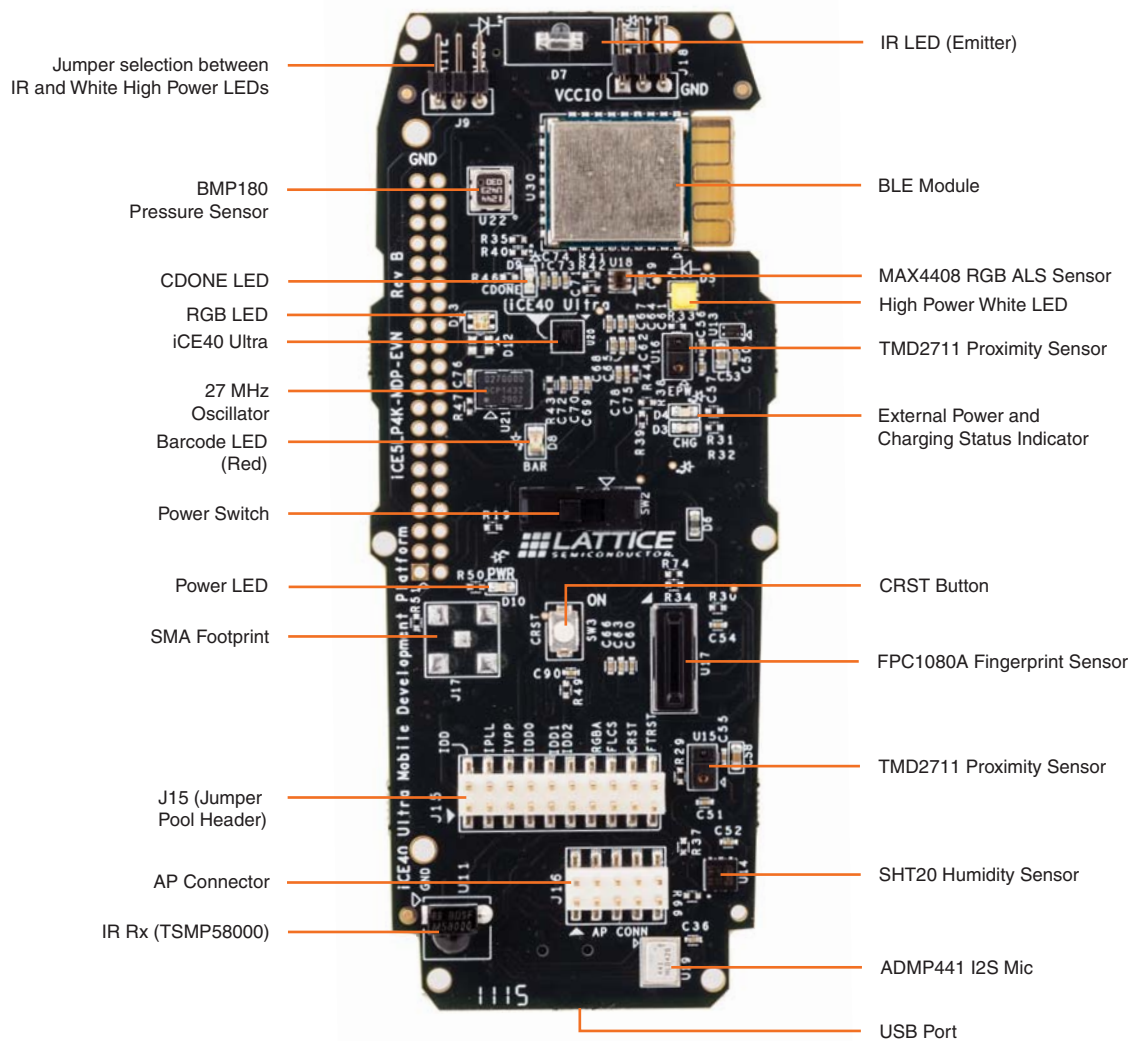
Figure 5. IR Self-Learning Demo Setup



iCE40 Ultra Mobile Development Platform Details

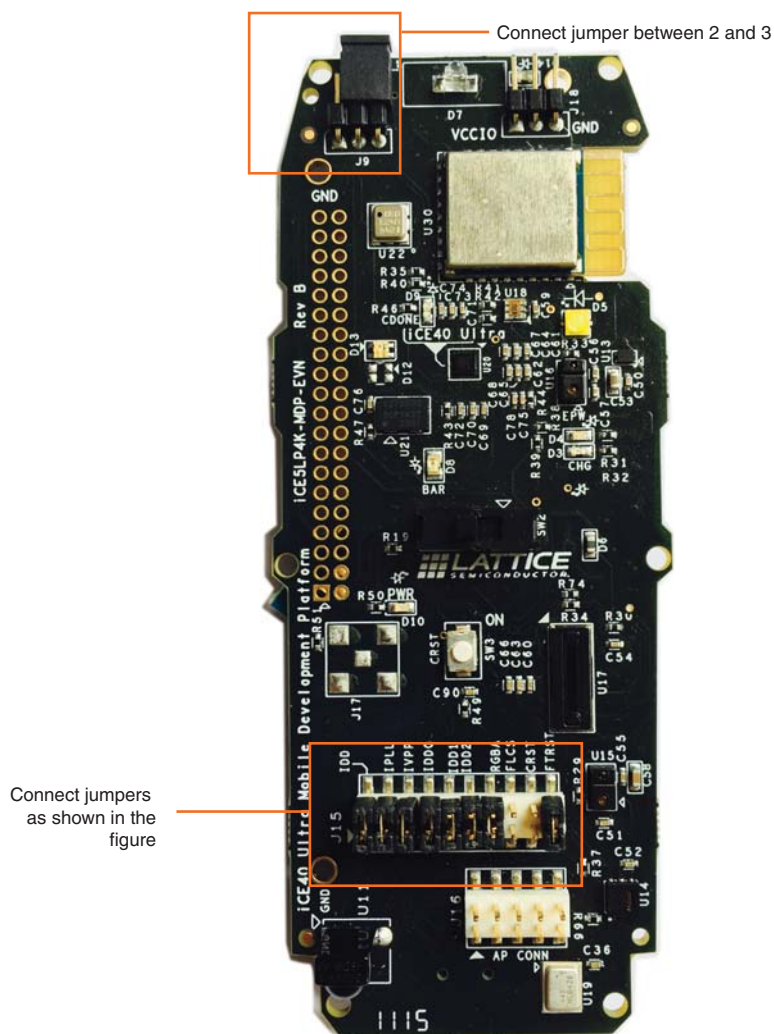
The details of the iCE40 Ultra Mobile Development Platform are shown in Figure 6.

Figure 6. iCE40 Ultra Mobile Development Platform Details



iCE40 Ultra Mobile Development Platform Default Jumper Settings

Figure 7. Default Jumper Settings



Set Key A or Key B to select an IR_IN (IR Rx). In the jumper pool J15, set all jumpers except FLCS and CRST. Set the jumper J9 to select IR LED as shown in the [iCE40 Ultra Mobile Development Platform Details](#) section.

Self-Learning IR Remote Controller Software Setup

The procedure for installing the Ble_Peel.apk to Android is described below.

To install BleIRSelfLearningDemo.apk to Android:

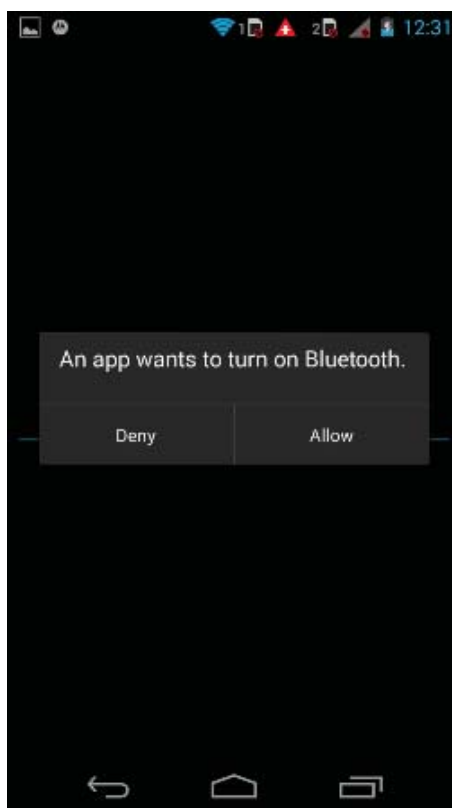
1. In the Android phone go to **Settings > Security** and select **Unknown sources**.
2. Copy BLE_Peel.apk from the *UG104\ demonstration\ Demo_apk* folder to the Android phone.
3. Go to File Manager and click the **BLE_Peel.apk** to install the application.
4. Deselect **Unknown sources**.

Connecting the iCE40 Ultra Mobile Development Board to the Android Phone

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

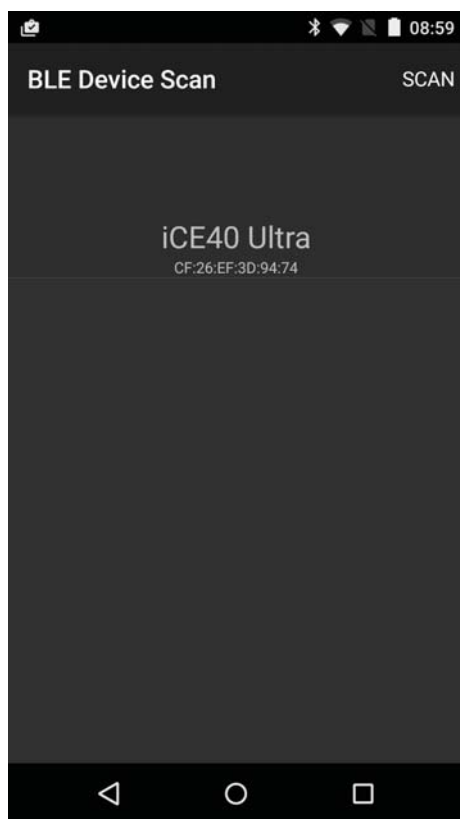
1. Turn on the mobile development board using switch SW2.
2. If Bluetooth is already enabled on the mobile, go to Settings and disable it.
3. Open the installed application by going to the Apps menu and selecting the **IR Self Learning demo**.
4. A pop-up window appears asking for permission to activate Bluetooth on the phone. Press **Allow** to activate Bluetooth.

Figure 8. Activate Bluetooth



5. A window appears that scans and lists BLE devices in the vicinity. Select the desired device to connect it to the phone.

Figure 9. 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.

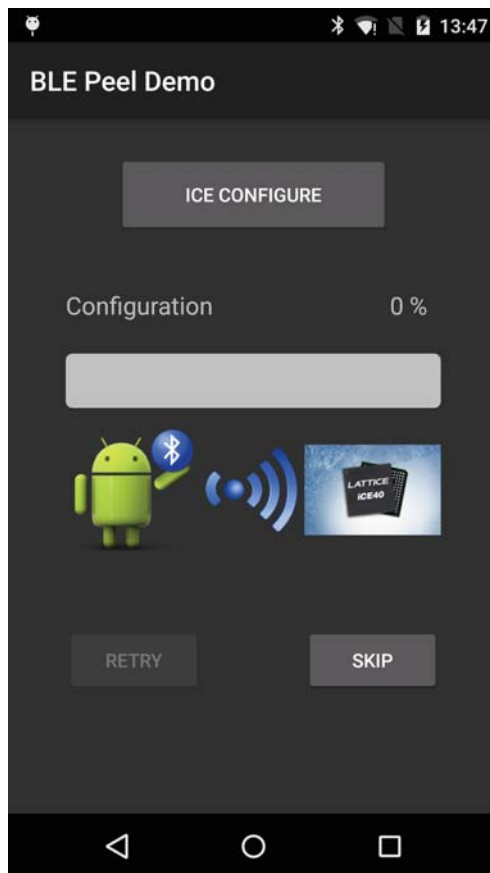
Configuring the CRAM Over-The-Air On the Android Phone

To Configure the CRAM Over-The-Air On the Android Phone

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

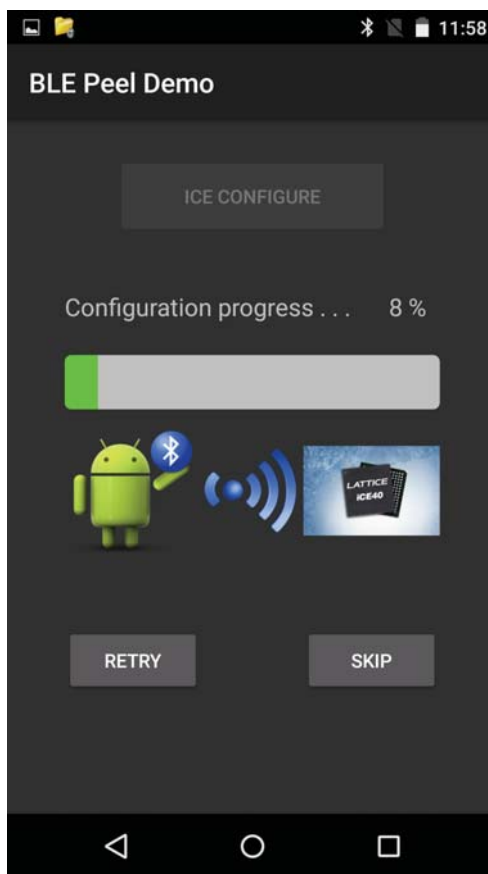
1. Once you connect to the iCE40 Ultra device, the following screen is displayed.

Figure 10. CRAM Configuration



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

Figure 11. Configuration Progress



When configuration is completed, the main application screen is displayed as shown in Figure 12.

Configuration Results

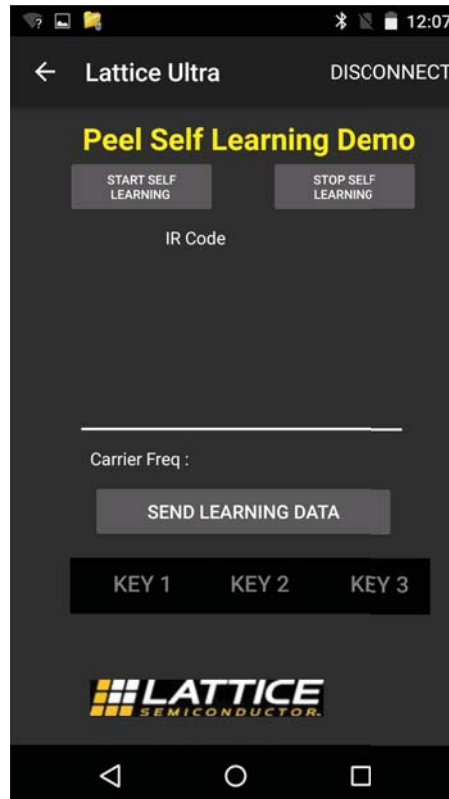
The average CRAM configuration time is 35 seconds.

Demo Procedure

To run the demo:

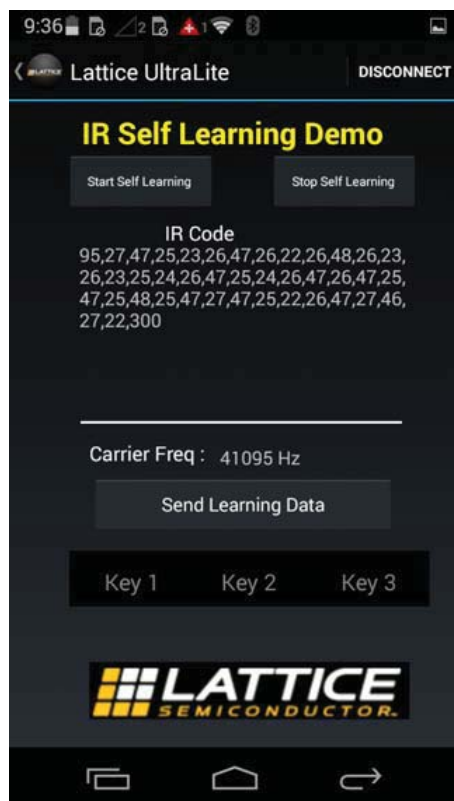
1. When the CDONE LED on the iCE40 Ultra board glows, unlock the screen if it is locked.
2. Go to the BLE Peel Self-Learning Demo application window as shown in Figure 12.

Figure 12. IR Self Learning BLE Demo Application



3. Press the **Start Self Learning** button. The application waits for the button to be pressed in the remote controller.
4. Press the button in the remote controller. The application reads the IR code from iCE40 Ultra through BLE and displays it under the field IR code. The IR carrier frequency is displayed in the Carrier Freq: text box as shown in Figure 13.

Figure 13. IR Code Display

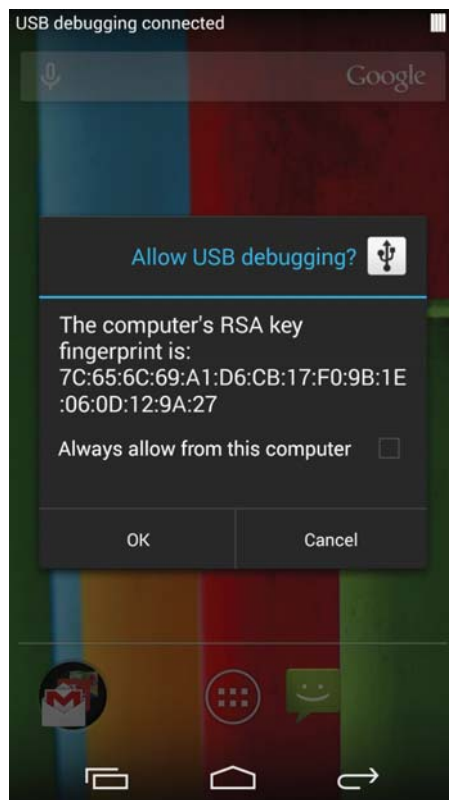


5. Press the **Stop Self Learning** button in the application. This terminates the self-learning mode.
6. Press the **Send Learning Data** button in the application. This send the IR code received during the self-learning period to the iCE40 Mobile Development Platform for transmission.
7. Verify the transmission by using the Sony Network Media Player. Place the Sony Network Media Player in front of the iCE40 Ultra such that IR Transmitter on it faces the front panel of the Sony Network Media Player.

Troubleshooting

When the mobile phone is connected to the host system, USB debugging is allowed as shown in Figure 14.

Figure 14. USB Debugging



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

1. Close the BLE Peel Demo apk process running in background.
2. In the Android menu, select **System settings > Applications > Manage Applications > IR Self Learning Demo > Force Stop**.
3. Open the IR Self Learning Demo application from the Android menu. The application is ready to be used as IR Self Learning Transceiver.

References

- DS1048, [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.

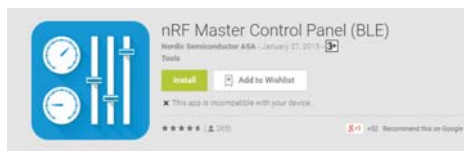
Appendix A. Steps to Program the BLE Module Over-The-Air

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 15. nRF Master Control Panel



On the iCE40 Ultra

To enter OTA mode:

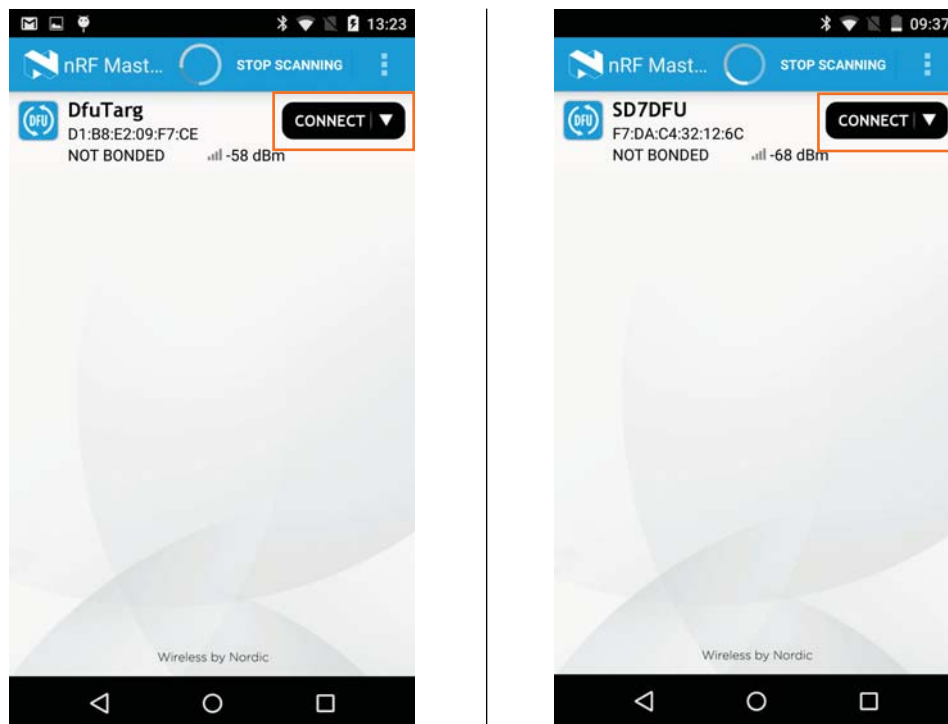
1. Join pins 2-3 on J18 (that is, connect P18 of the BLE Module to GND).
2. Power on the iCE40 Ultra board.
3. Open the installed **nRF Master Control Panel** application.

As shown in Figure 16, 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.

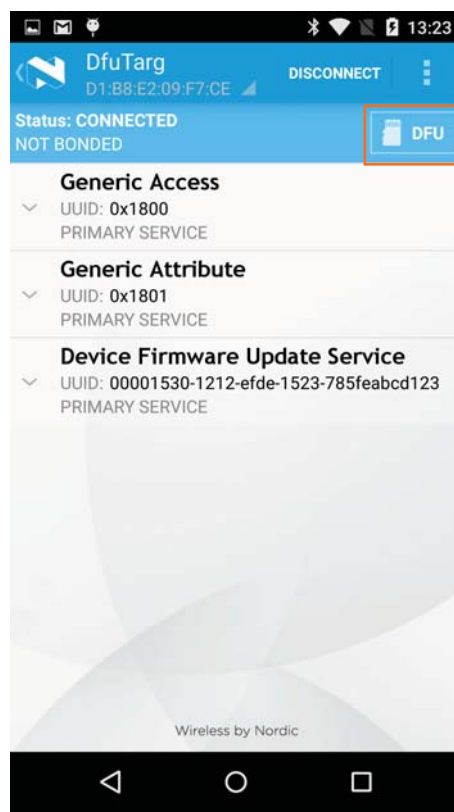
Figure 16. .DfuTarg/SD7DFU BLE Device BLE Device



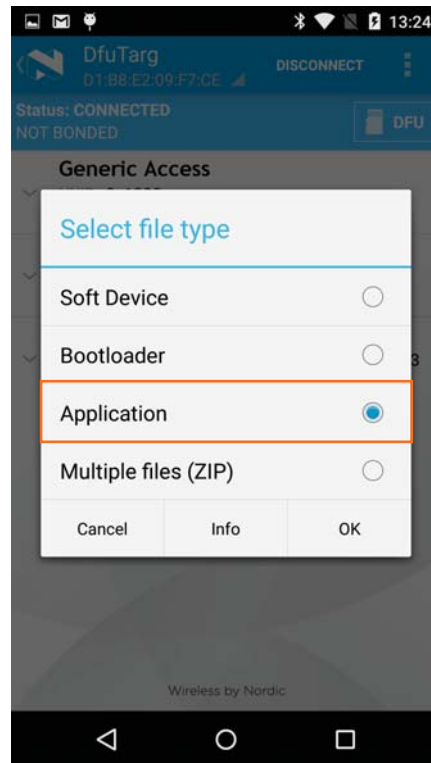
4. Press **CONNECT** to open the window shown in Figure 17.

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

Figure 17. DFU Option



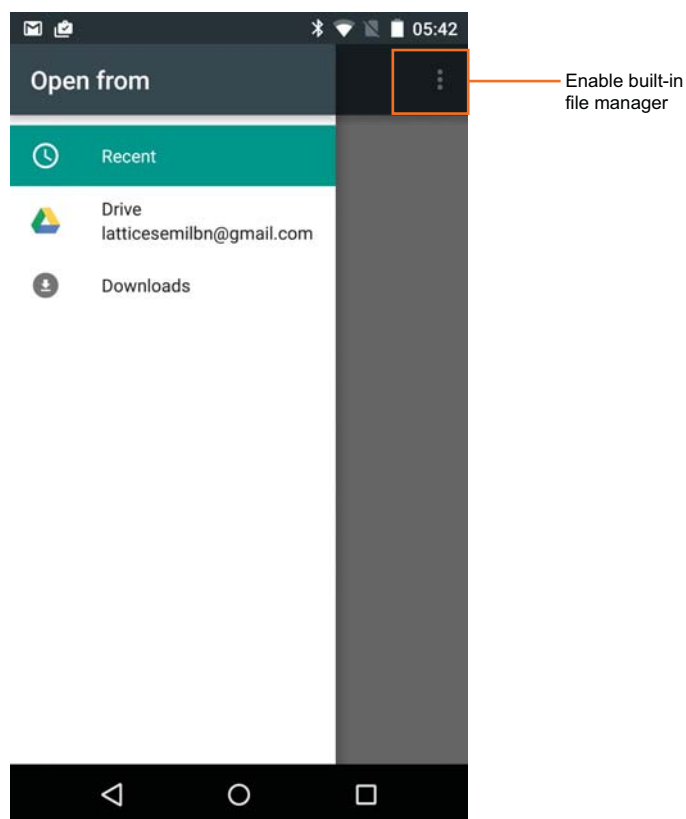
5. Press the **DFU** option.
6. The resulting window, after connection is established, is shown in Figure 18. Select the file type **Application**.
7. Press **OK**.

Figure 18. File Type Selection

8. Set up the file manager.

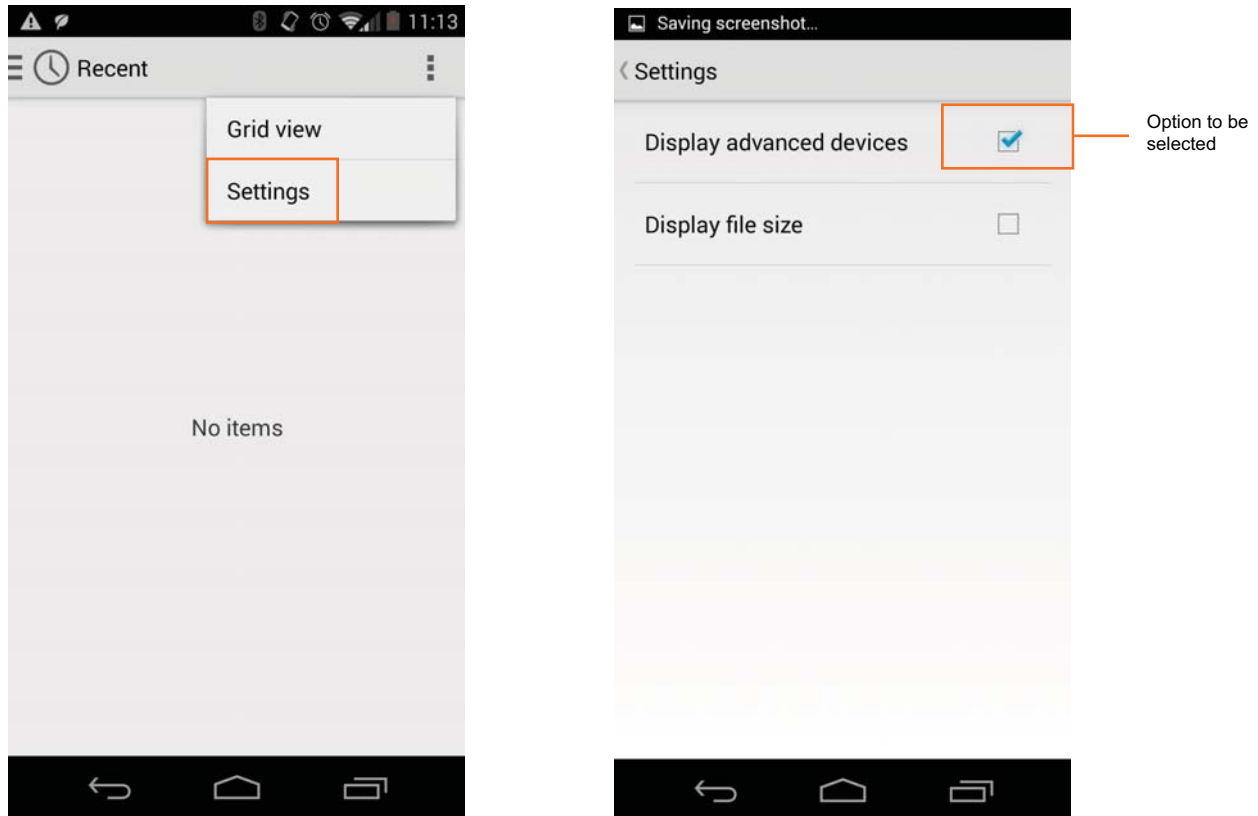
If a file manager is already installed, proceed to Step 9.

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 19.

Figure 19. File Manager Selection Window

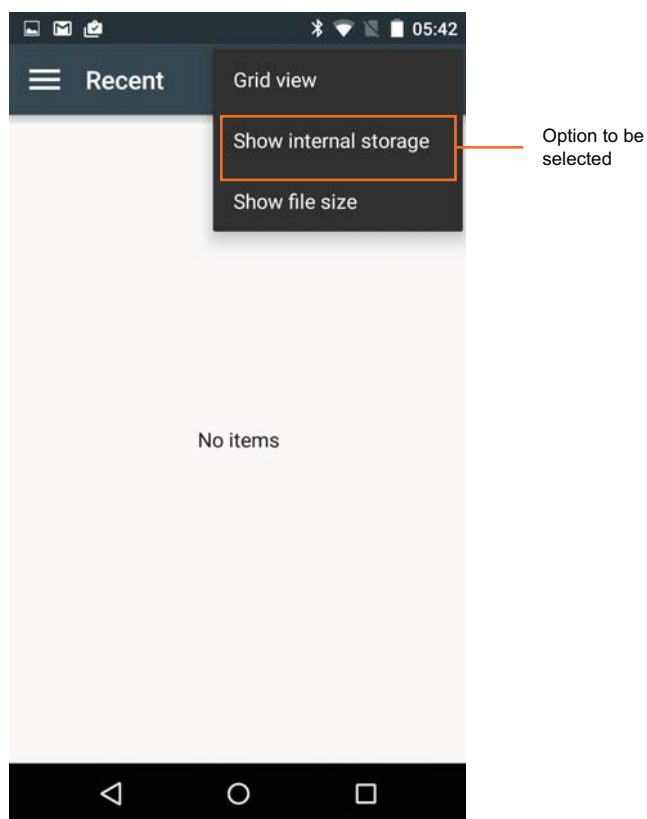
In a KitKat operating system, after pressing the top-right settings button, the resulting window is shown in Figure 20.

Figure 20. Option for KitKat 4.4 OS



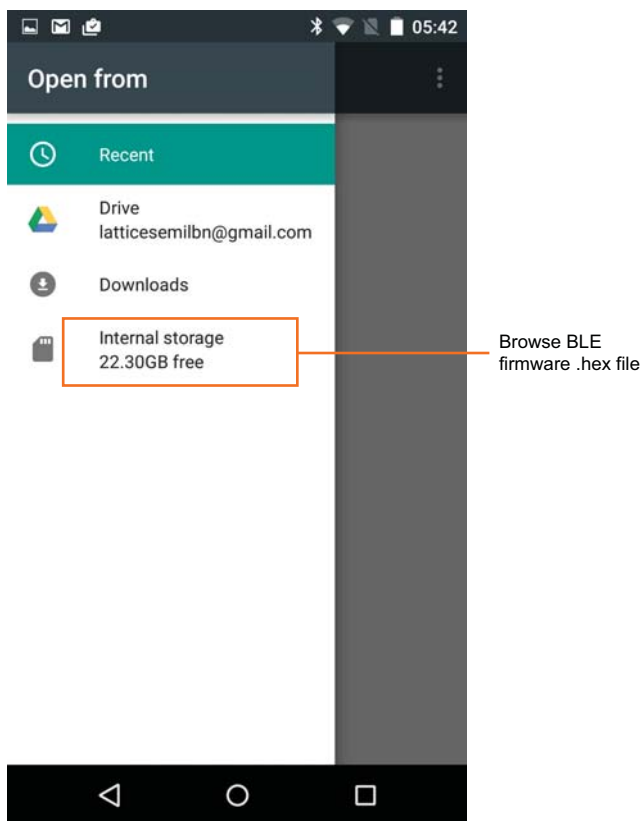
In a Lollipop operating system, the resulting window is shown in Figure 21.

Figure 21. Option for Lollipop OS



After following the above procedure, the resulting window is shown in the Figure 22.

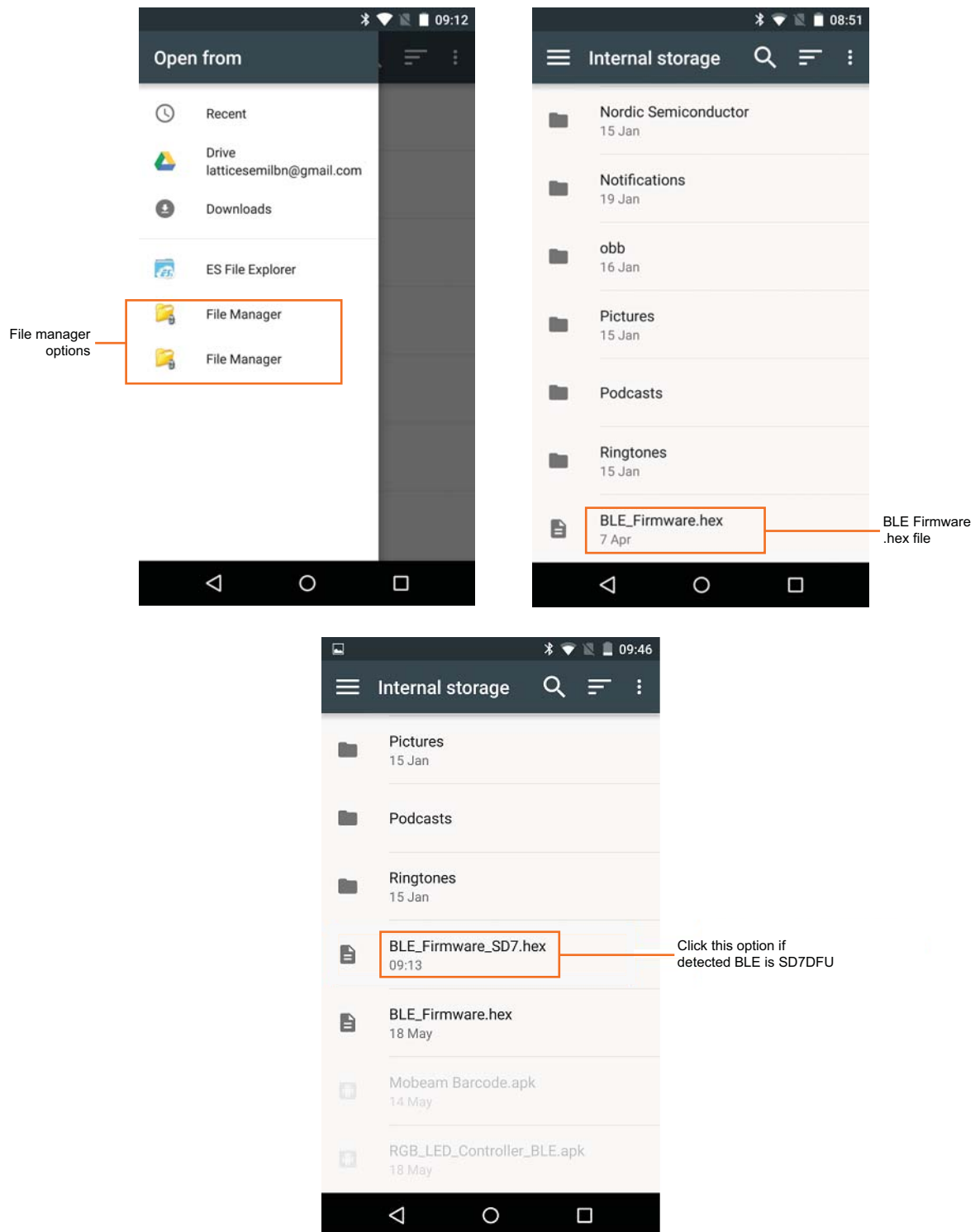
Figure 22. Enabled Built-in File Manager



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

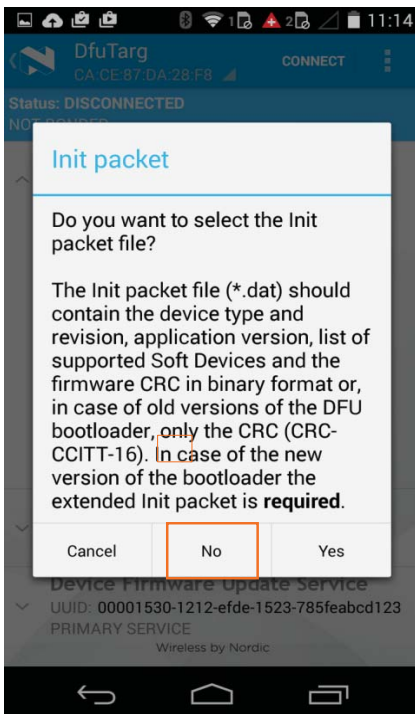
As shown in Figure 23, choose **BLE_Firmware.hex** if the device scanned is *DfuTarg* (Nordic S110 SoftDevice rev 6.0). Choose **BLE_Firmware_SD7DFU.hex** if the device scanned is *SD7DFU* (Nordic S110 SoftDevice rev 7.0).

Figure 23. BLE Firmware .hex File



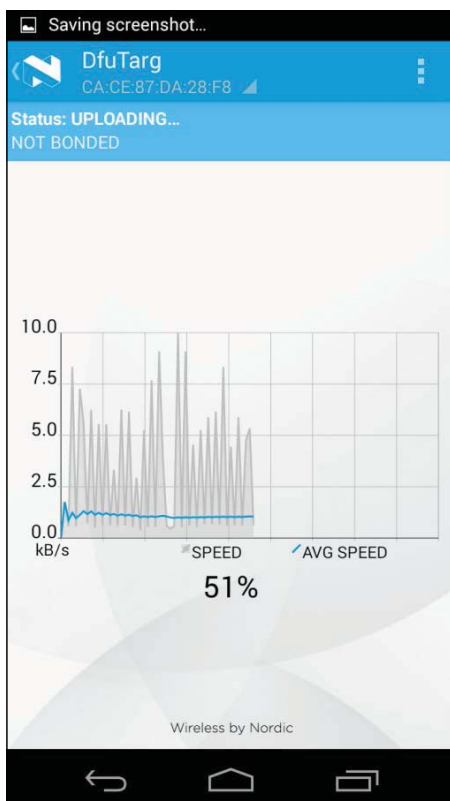
10. Select **No** as shown in Figure 24.

Figure 24. Init Packet



11. The upload progress is displayed on the interface.

Figure 25. 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.