



## iCE40 Ultra™ Self-Learning IR Remote

---

**User's Guide**

## Introduction

This guide describes how to use the iCE40 Ultra™ Mobile Development Platform for demonstrating the Self-Learning IR Remote Controller design for user application. 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.

The document discusses complete demonstration steps and the associated designs.

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.
- Utilize the additional hardware required to run the demonstrations.
- Understand the design details of the Self-Learning IR Remote Controller demo implemented on iCE40 Ultra.
- Run the demo along with the Intrinsic® DragonBoard™.
- 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, refer to:

- EB90, [iCE40 Ultra Mobile Development Platform User's Guide](#)
- DS1048, [iCE40 Ultra Family Data Sheet](#)

This document is divided into two sections. The first section describes the Self-Learning IR Remote Controller demonstration in detail and the second section describes the Self-Learning IR Remote Controller design. The Self-Learning IR Remote Controller demonstration is performed using either an I2C or an SPI interface with the Processor.

## Self-Learning IR Remote Controller Demonstration

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

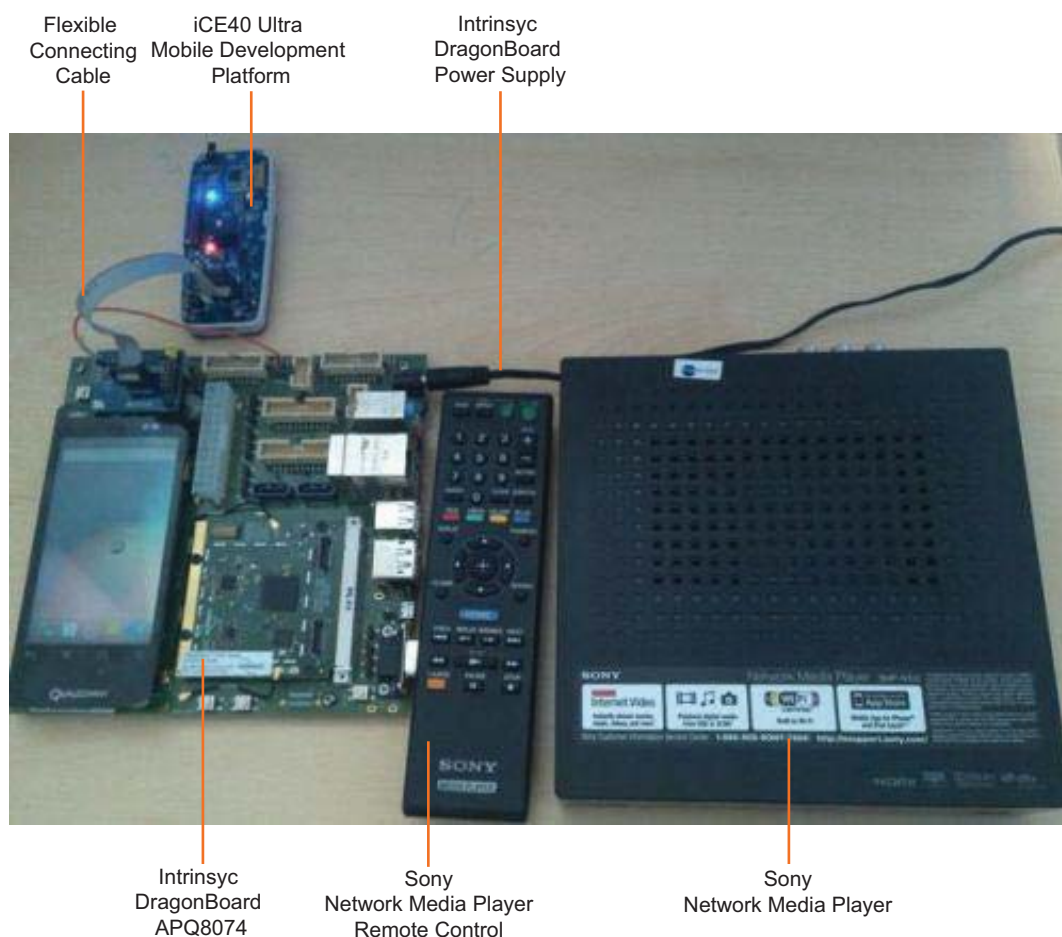
### Self-Learning IR Remote Controller Demonstration Setup Using SPI Interface

The Self-Learning IR Remote Controller demonstration setup using SPI interface consists of the following components:

- APQ8074 Intrinsic DragonBoard
- iCE40 Ultra Mobile Development Platform
- Sony® Network Media Player
- Sony Network Media Player remote and flexible connecting cables

Connect the iCE40 Ultra Mobile Development Platform to the adapter board mounted on the Intrinsic APQ8074 DragonBoard as shown in Figure 1. The sensor header is directly above the display.

**Figure 1. Connecting the iCE40 Ultra Mobile Development Platform - SPI Interface**



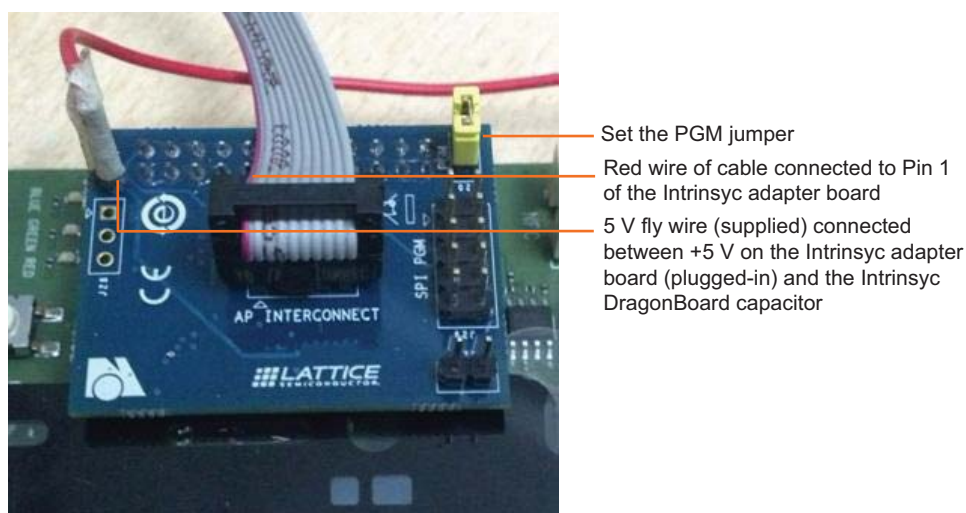
## Connecting the iCE40 Ultra Mobile Development Platform to the Intrinsic DragonBoard

To connect the iCE40 Ultra Mobile Development Platform to the Intrinsic DragonBoard:

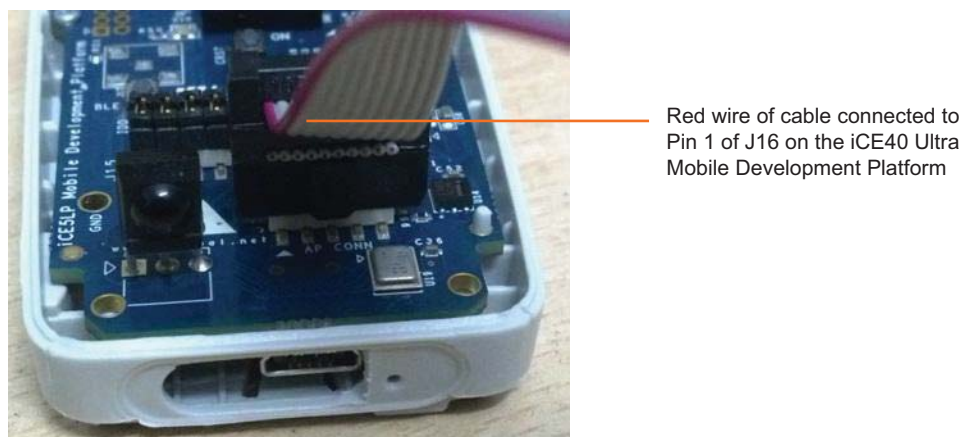
1. Power-off the Intrinsic DragonBoard.
2. Connect one end of the flexible connecting cable to the adapter board mounted on the Intrinsic DragonBoard. The red wire of the cable connector should be connected to Pin 1 of the connector on the adapter board. Pin 1 is located near the white triangle.
3. Connect the other end of the cable to the iCE40 Ultra Mobile Development Platform. The red wire of the cable connector should be connected to Pin 1 of the J16 connector. Pin 1 is located near the white triangle.
4. Power-on the Intrinsic DragonBoard.

The connections are shown in Figure 2 and Figure 3.

**Figure 2. Cable Connection to Adapter Board**



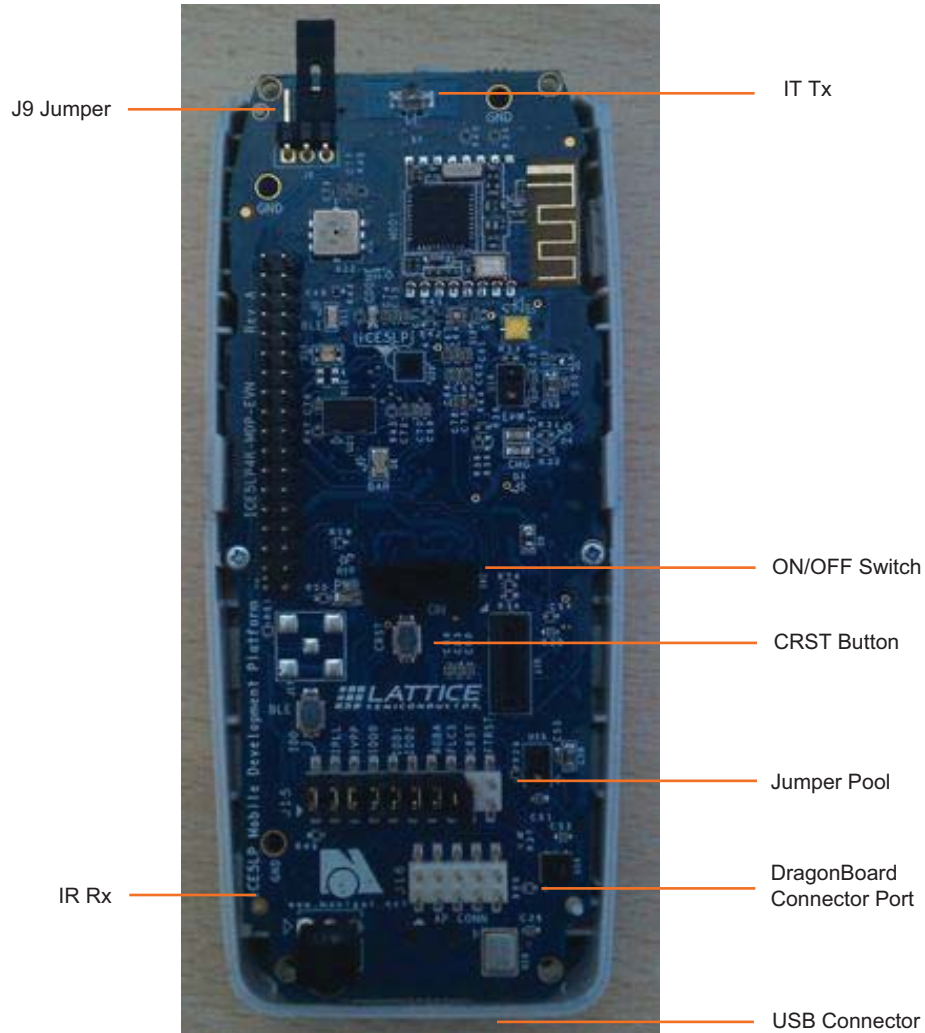
**Figure 3. Cable Connection to iCE40 Ultra Board**



## iCE40 Ultra Mobile Development Platform Details

The details of the iCE40 Ultra Mobile Development Platform are shown in Figure 4.

**Figure 4. iCE40 Ultra Mobile Development Platform Details**



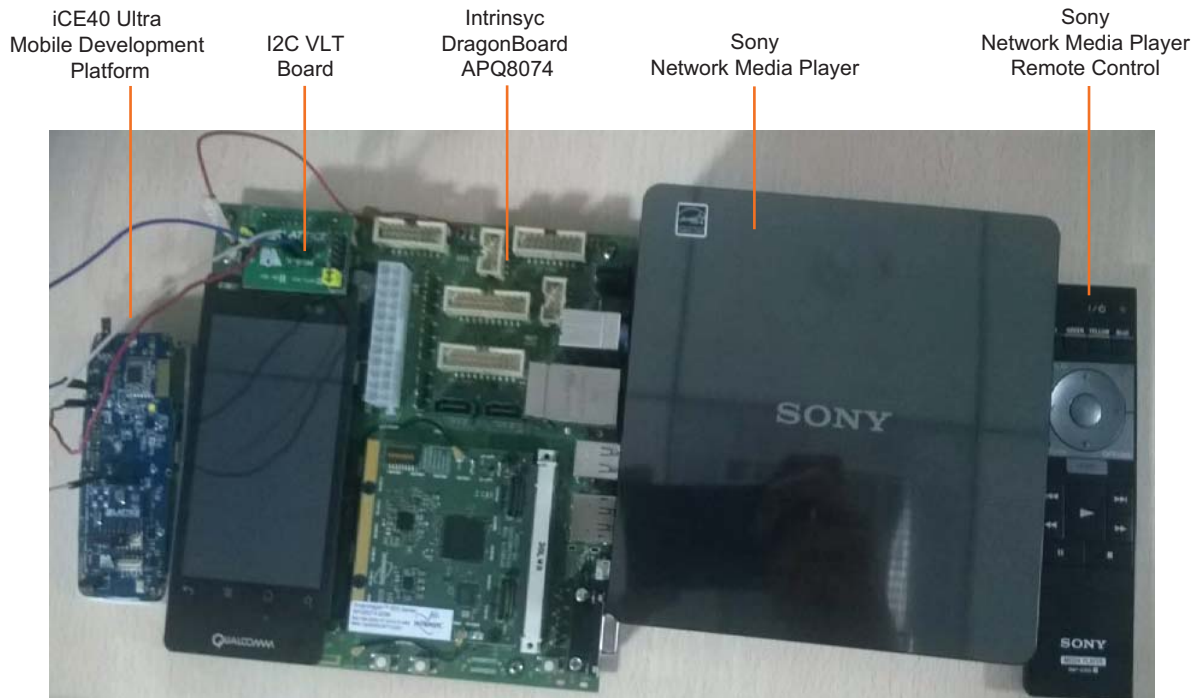
## Self-Learning IR Remote Controller Demonstration Setup Using I2C Interface

The Self-Learning IR Remote Controller demonstration setup using I2C interface consists of the following components:

- APQ8074 Intrinsic DragonBoard Gen 2 with +5V adaptor and USB debugger cable
- iCE40 Ultra Mobile Development Platform with programming USB cable
- I2C VLT board
- Sony® Network Media Player
- Sony Network Media Player remote
- Three flexible connecting cables

Connect the iCE40 Ultra Mobile Development Platform to the i2C VLT board mounted on the Intrinsic APQ8074 DragonBoard as shown in Figure 5.

**Figure 5. Connecting the iCE40 Ultra Mobile Development Platform - I2C Interface**



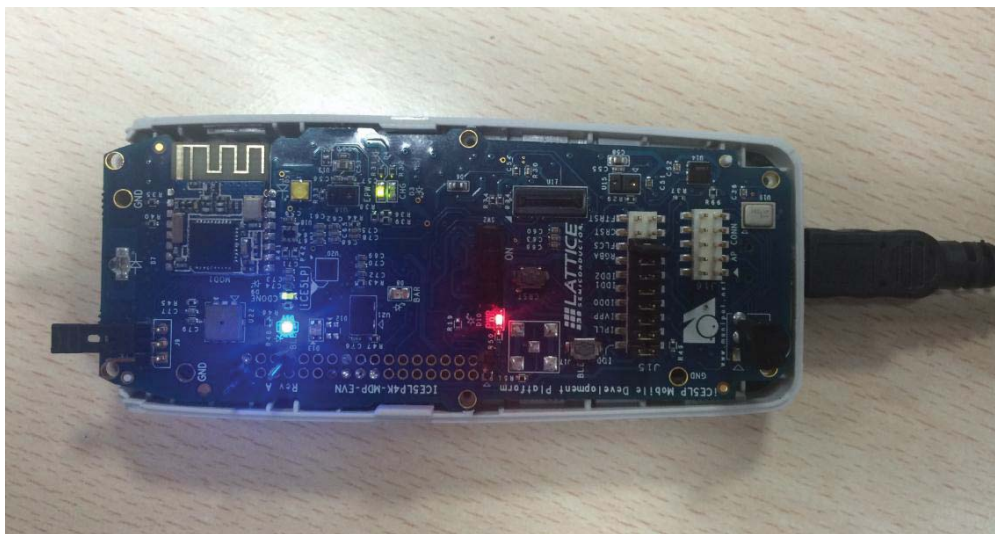


## Programming the iCE40 Ultra Mobile Development Platform

To program the iCE40 Mobile Development Platform:

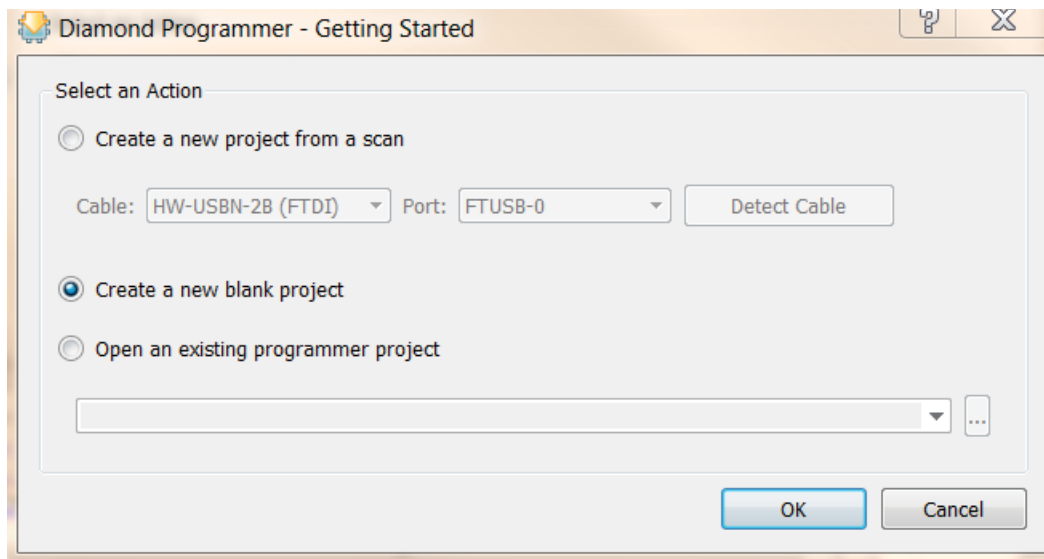
1. Connect the iCE40 Mobile Development Platform to the USB port of the PC.

**Figure 6. Connecting Board to PC**



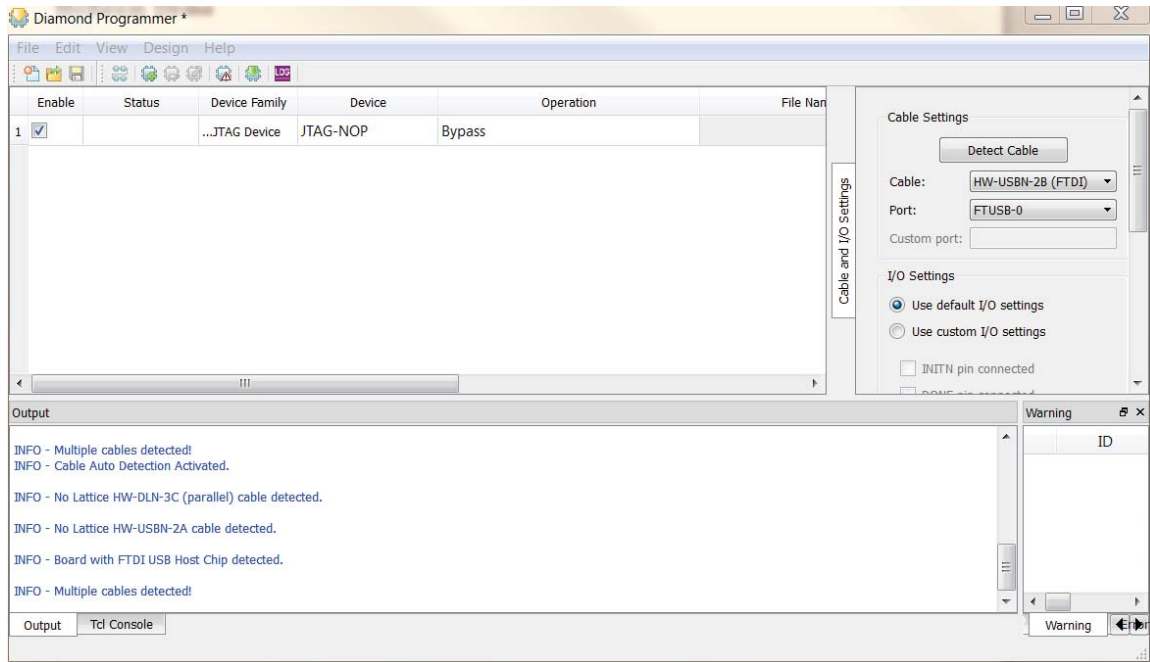
2. Open the Diamond® Programmer version 3.2 and above.
3. In the Getting Started dialog box, select **Create a new blank project** and click **OK**.

**Figure 7. Creating New Project**



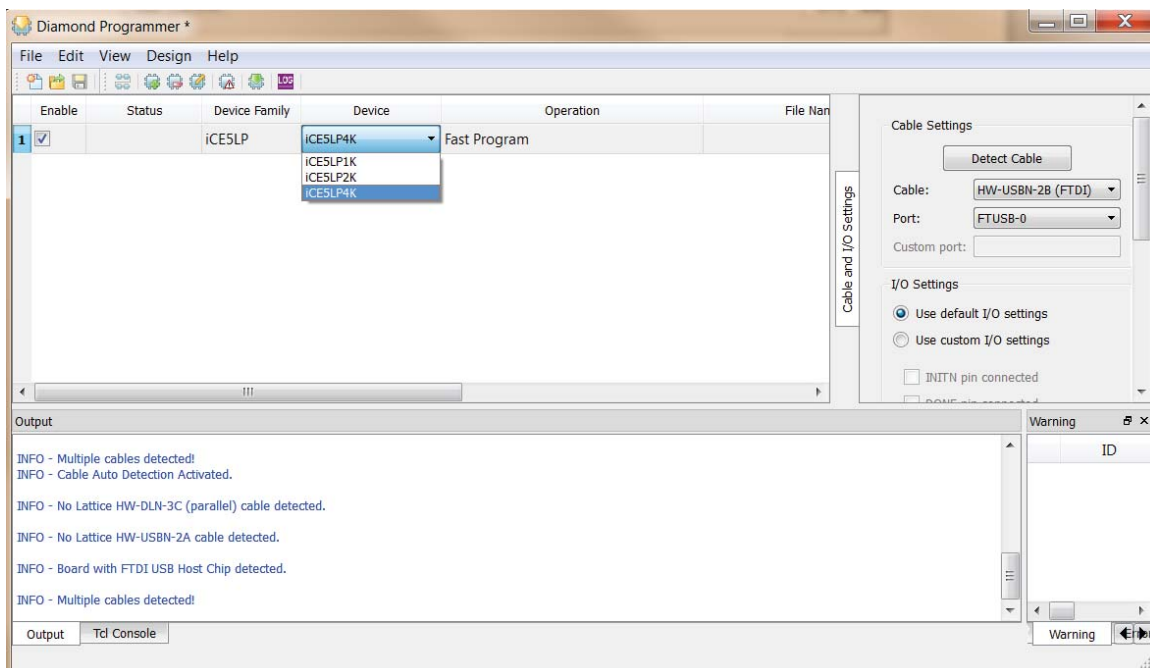
This opens the Diamond Programmer main interface.

**Figure 8. Diamond Programmer Interface**



4. Select **iCE5LP** under Device Family and then select **iCE5LP4K** under Device.

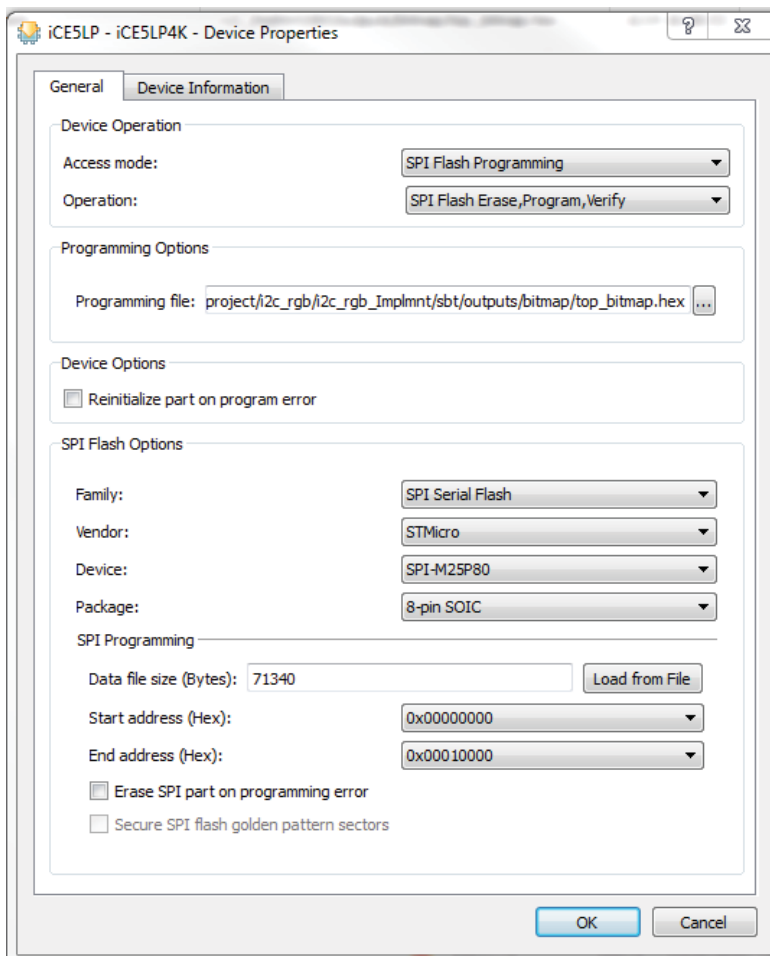
**Figure 9. Selecting the Device**





5. Double-click on Fast Program and the .bin file provided in the demonstration `/bitstream/i2c` folder.

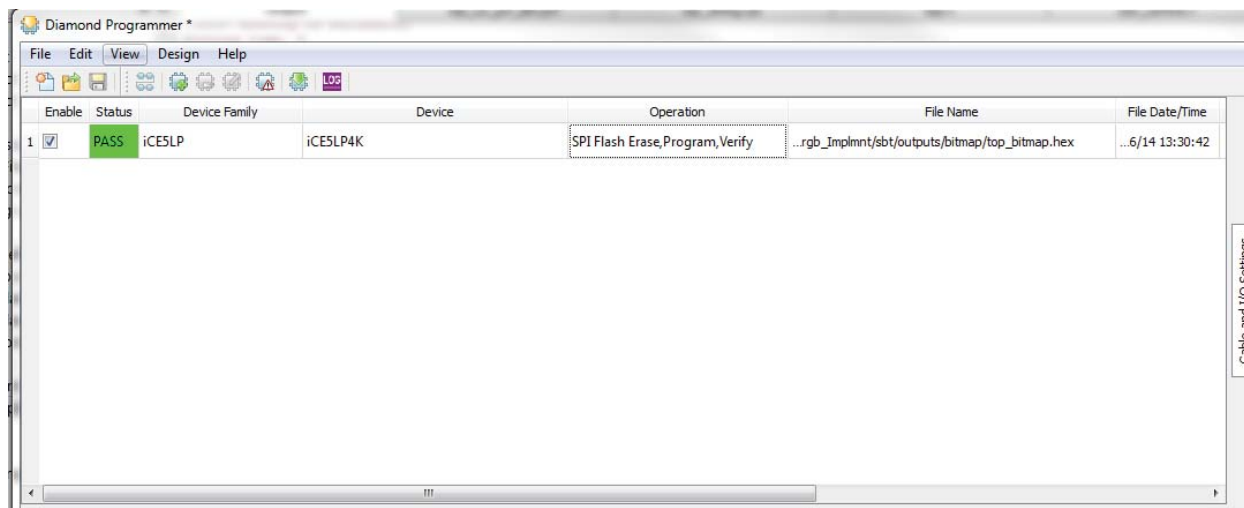
**Figure 10. Selecting Programming File**



6. Select the program to use in programming the device.

7. Verify that the operation has successfully completed.

**Figure 11. Verifying Operation**



## Connecting iCE40 Ultra Mobile Development Platform to SnapDragon Board APQ8074

To connect iCE40 Ultra Mobile Development Platform to SnapDragon Board APQ8074:

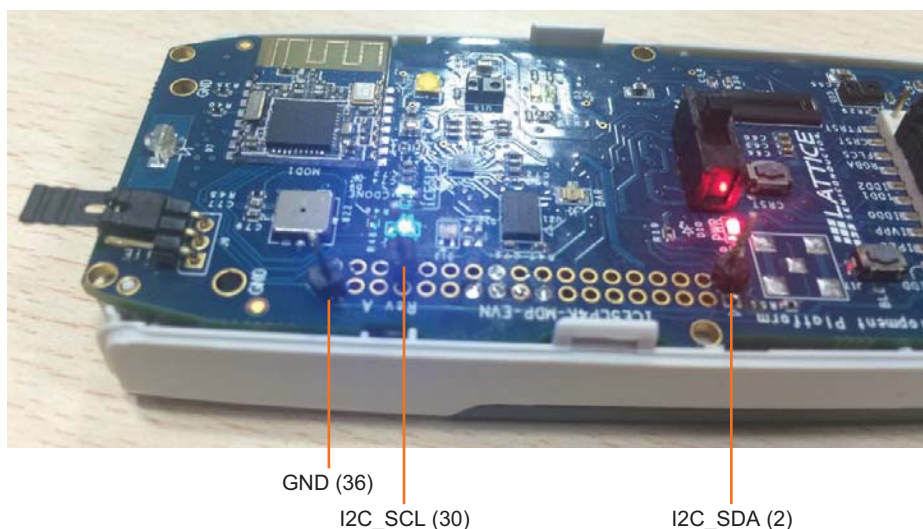
1. Connect the I2C lines between I2C VLT board and iCE40 Ultra Mobile Development Platform as shown in Table 1.

**Table 1. I2C Line Connections**

Sr. No.	I2C VLT Board [J6 Jumper] Pin Numbers	Signal	iCE40 Ultra Mobile Development Platform [J10 Jumper] Pin Numbers
1	4	I2C_SDA	2
2	3	I2C_SCL	30
3	9	Ground	18 / 36

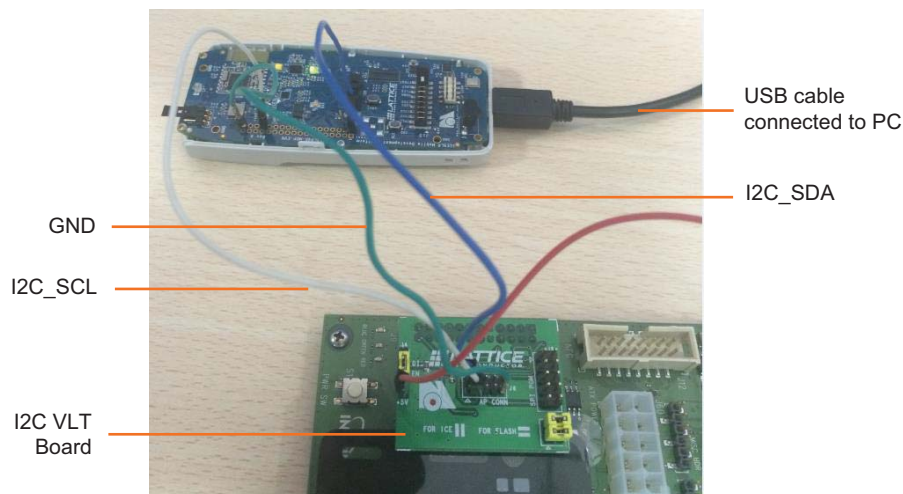
2. Solder three jumper pins as shown in the Figure 12.

**Figure 12. Soldering Jumper Pins**



3. Connect the I2C lines as shown in Figure 13.

**Figure 13. Connecting I2C Lines**



---

## **iCE40 Ultra Mobile Development Platform 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.

Note: For processor configuration demo, set FTRST. Do not set FLCS.

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**

This section provides the procedures in downloading Flash system and boot images to Intrinsic DragonBoard.

*Note: This procedure is not required if the DragonBoard is already flashed with the system and boot images.*

To flash system image and boot image to Intrinsic DragonBoard:

1. Make OTG (mini-USB) connection from Intrinsic DragonBoard mini USB port of your Host system.
2. Download and install android-sdk. Android-sdk is available for Linux and Windows environments from <http://developer.android.com/sdk/index.html>.

*Note: Add the installation location /android\_sdk/platform-tools/ in your system PATH variable.*

3. Run the command below in the command prompt.

```
#sudo -s
```

*Note: This command is only applicable for Linux machines. In Windows, administrative permission is required.*

4. Reboot the Intrinsic DragonBoard in FASTBOOT mode.

Keep the S2 button pressed on the Intrinsic DragonBoard during power on. If the board is already powered ON and is in adb mode, run the command below for FASTBOOT mode.

```
#adb reboot bootloader
```

5. When the Intrinsic DragonBoard is in FASTBOOT mode, a white screen is displayed with only the Intrinsic name.

Run the command below in the command prompt. The FASTBOOT device number and its name is listed.

```
#fastboot devices
```

When the procedure is completed, the board is ready to be flashed with the system and boot images.

6. Download APQ8074\_JB\_BootImage.zip and extract the contents below.

For I2C: /APQ8074\_JB\_BootImage/I2C\_img

For SPI: /APQ8074\_JB\_BootImage/SPI\_img

7. To flash the system and boot images, run the script file *flashall.sh*.

For I2C: From the /APQ8074\_JB\_BootImage/I2C\_img directory, run the script file *flashall.sh*.

For SPI: From the / APQ8074\_JB\_BootImage /SPI\_img directory, run the script file *flashall.sh*.

Alternatively, you can run the commands below to flash the system image.

For I2C:

```
#cd APQ8074_JB_BootImage /I2C_img
#fastboot flash system system.img
```

For SPI:

```
#cd APQ8074_JB_BootImage /SPI_img
#fastboot flash system system.img
```

If flashing is successful, OKAY and Finished are displayed on the terminal.

Run the command below to flash the boot image.

```
#fastboot flash boot boot.img
```

If flashing is successful, OKAY and Finished are displayed on the terminal.

8. Reboot the board to verify the current board images using the command below.  

```
#fastboot reboot
```
9. After reboot is completed, go to System settings > About phone. Scroll down and tap on **Build number** seven times to enable Developer options.
10. Go to Developer options and select the **Stay awake** check box to keep the DragonBoard awake at all times.

## Installing Peel.apk to Android

To install Peel.apk to Android:

1. Make OTG (mini-USB) connection from the Intrinsic DragonBoard mini USB port of your Host system.
2. Run the command below on the command prompt.

```
#sudo -s
```

3. To establish and verify adb connection, run the commands below.

```
#adb kill-server
#adb start-server
#adb devices
```

If the connection is successful, the device ID is displayed on the terminal.

4. To install *Peel.apk* to the Intrinsic Dragonboard, run the command below.

```
cd demonstration/Demo_Apk/
#adb root
#adb remount
#adb install Peel.apk
```

- The application searches for the FPGA bitmap in the `/etc/firmware/` directory. Manually push the FPGA bitmap to this location after installing the application. To do this, run the command below.

For SPI:

```
#cd demonstration /bitstream/spi
#adb push peel_bitmap.bin /etc/firmware
#adb push peel_config /system/etc
```

For I2C:

```
#adb push peel_config /system/etc
#cd demonstration/bitstream/i2c/peel_bitmap.bin
```

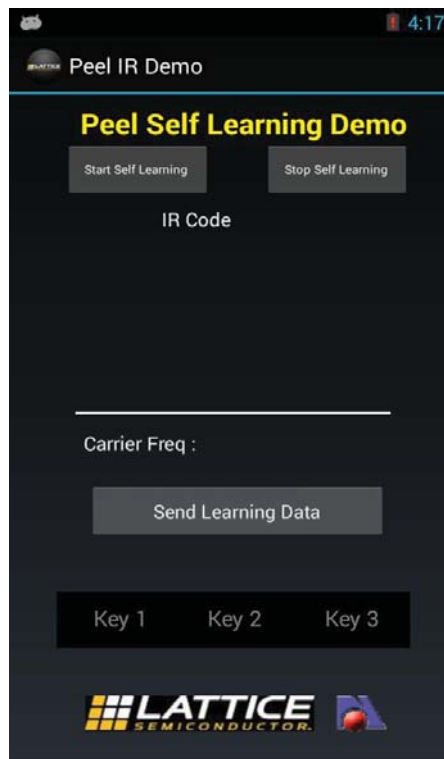
Use the bin file to program iCE40 Ultra device.

## Demo Procedure

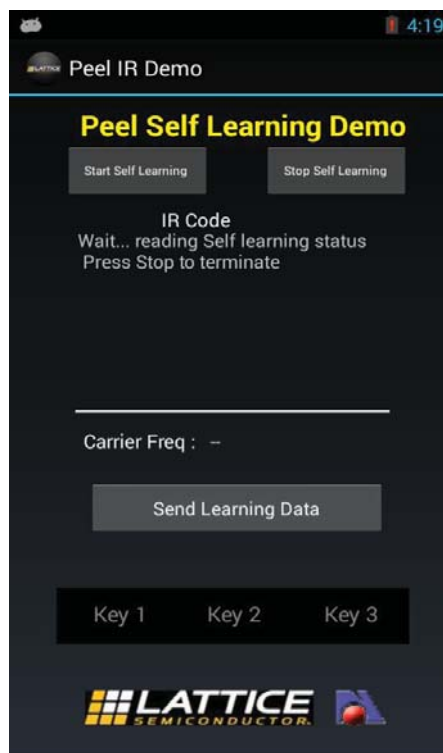
To run the demo:

- Connect the iCE40 Ultra Mobile Development Platform to DragonBoard using the flexible connecting cable.
- Power ON Dragonboard and wait for the boot sequence to complete and the Home screen to appear.
- Unlock the screen. Go to the Android application menu and click the **Peel IR Demo** application.
- Wait for Processor Configuration to be completed, which is indicated by the glowing of the CDONE LED on iCE40 Ultra Mobile Development Platform. The application is now ready to use as a Peel IR transceiver as shown in Figure 14.

**Figure 14. Self-Learning IR Remote Demo Application**



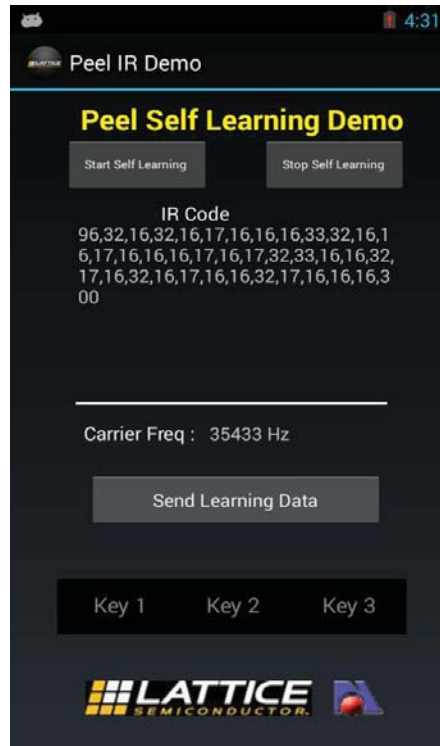
- In the application click **Start Self Learning** button. The status is displayed as shown in Figure 15.

**Figure 15. Process Status**

The application waits for the button to be clicked in the remote controller.

6. Click the button in the remote controller. The application reads the IR code from iCE40 Ultra through SPI and displays it under the field IR code. The IR carrier frequency is displayed in the Carrier Freq; text box as shown in Figure 16.



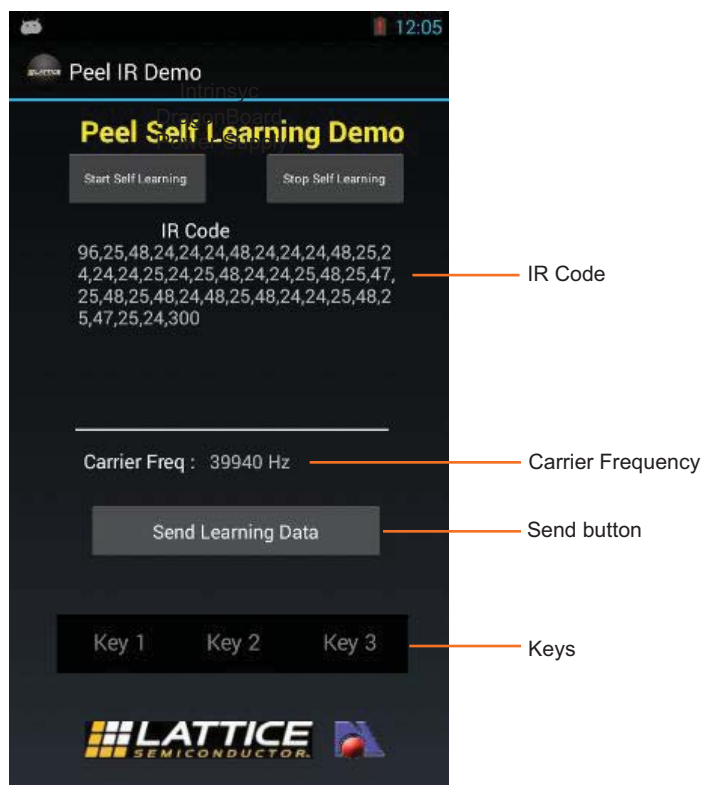
**Figure 16. IR Code Display**

7. Press the **Stop Self Learning** button in the application. This terminates the self learning mode.
8. 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.
9. 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.

## Self-Learning IR Remote SPI Demo Application Features

The Self-Learning IR Remote SPI Demo application features is shown in Figure 17.

**Figure 17. Self-Learning IR Remote SPI Demo Application Features**



- **Start Self Learning** button prompts the application to read the learning status, the corresponding IR carrier frequency and the IR code from the iCE40 Ultra Mobile Development Platform.
- Pressing the **Stop Self Learning** button terminates the self learning process.
- IR code contains the decoded code for the corresponding button pressed on the remote.
- Carrier frequency represents frequency of IR remote.
- The **Send Learning Data** button transmits the learned data to the iCE40 Ultra Mobile Development Platform.
- Key1, Key2 and Key3 are used as test keys.

## Troubleshooting

If the Android application does not respond, perform the following procedure:

1. Close the Self-Learning IR Remote Demo process running in background.
2. In the Android menu, select **System settings > Applications > Manage Applications > Peel Demo > Force Stop**.
3. Restart Intrinsic DragonBoard.
4. Open the Peel Demo application from the Android menu. The application is ready to use as Peel IR transceiver.

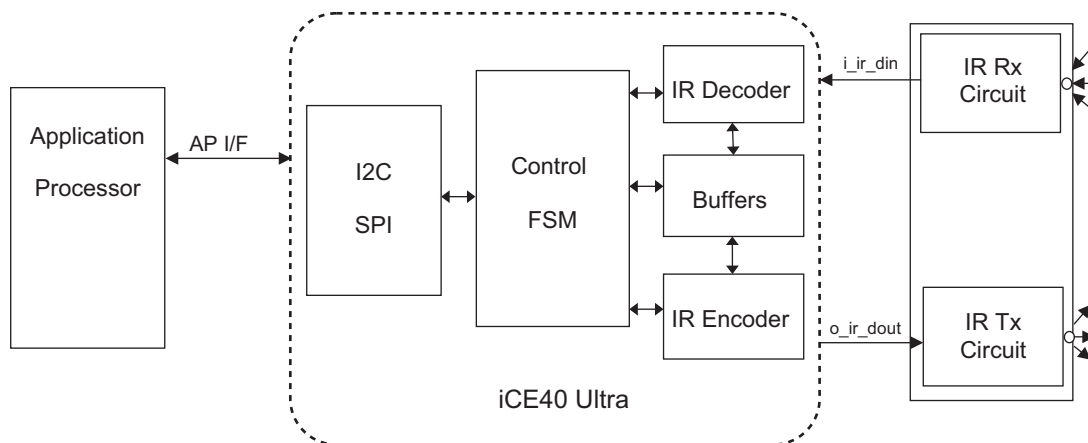
## Self-Learning IR Remote Controller Design

The following section describes the internal details of the Self-Learning IR Remote Controller design.

### Overview

This design example enables the capabilities of the Lattice iCE40 Ultra devices as a self-learning IR remote controller. Figure 18 shows the block diagram of data-base based self-learning remote controller with processor interface.

**Figure 18. Block Diagram**



### Design Features

- Multiple universal interfaces (I2C, and SPI).
- Configurable Carrier Frequency
- Self Learning Capability
- Interface: 16 bits Address, 8 bits Data

### I2C Interface and Write/Read Operation Features

- Slave Mode
- Device ID: 7'b1100001
- Supports single-byte or sequential read and write

**Figure 19. I2C Write Operation**

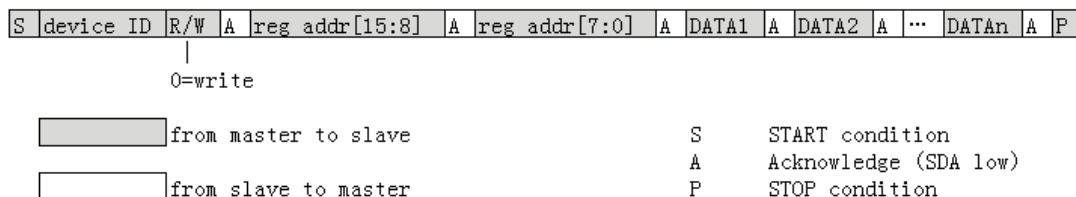
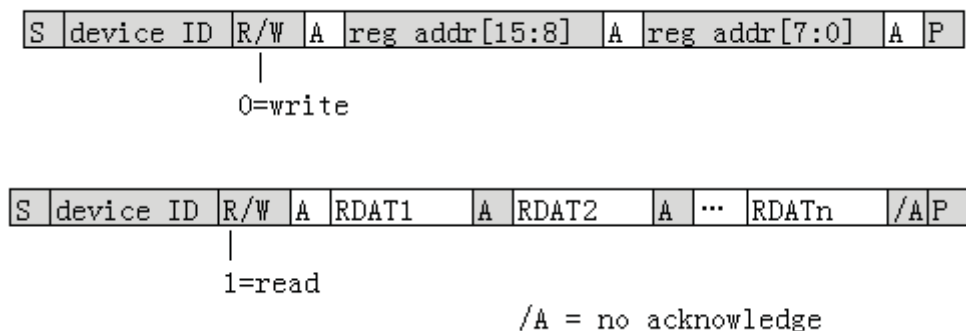


Figure 20. I2C Read Operation



### SPI Interface and Write/Read Operation Features

- Slave Mode
- CPOL=0, CPHA=1
- Supports single-byte or sequential read and write

Table 2. SPI Write Operation

MOSI Byte Index	MISO Byte Index	Description
1	n/a	0x02
2	n/a	Address High Byte, Address[15:8]
3	n/a	Address Low Byte, Address[7:0]
4	n/a	DATA 1
5	n/a	DATA 2
6	n/a	DATA 3
•	n/a	•
N+3	n/a	DATA N

Table 3. SPI Read Operation

MOSI Byte Index	MISO Byte Index	Description
1	n/a	0x03
2	n/a	Address High Byte, Address[15:8]
3	n/a	Address Low Byte, Address[7:0]
n/a	4	Read Data 1
n/a	5	Read Data 2
n/a	6	Read Data 3
n/a	•	•
n/a	N+3	Read Data N

## Registers Description

Table 4 provides the descriptions of the registers.

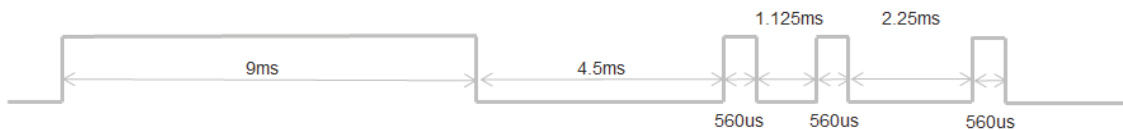
**Table 4. Registers Description**

ADDRESS	NAME	R/W	Description
0x0000	Command	R/W	Bit 0: Rising edge, starts transmit. Bit 1: Rising edge, starts self-learning.
0x0001	Status	R	Bit 0: High means self-learning data have been stored to Rx Buffer.
0x0002	Tx Length Low Byte	R/W	Valid data bytes in Tx buffer.
0x0003	Tx Length High Byte	R/W	
0x0004	Rx Length Low Byte	R	Valid data bytes in Rx buffer.
0x0005	Rx Length High Byte	R	
0x0006	Carrier Divider Low Byte	R/W	Write this register to determine the transmission carrier frequency. Read this register to get the carrier frequency after self-learning. Carrier Clock divider. Such as: System Clock: 27000000 Hz Carrier Frequency: 38400 Hz Divider = 27000000 / 38400 = 703 = 0x2BF
0x0007	Carrier Divider High Byte	R/W	
0x0008	System Clock Frequency	R	Optional to use. The unit is KHz. Such as: System Clock: 27000000 Hz Clock Frequency Low Byte = 0x78 Clock Frequency High Byte = 0x69
0x0009	System Clock Frequency High Byte	R	

## Buffer Description

Both of Tx and Rx Buffer size are 2048 bytes, and share the same address, from 0x0800 to 0xff. The TX/RX buffer include IR commands to describe a IR frame. For example, if the IR Command Values are 9000, 4500, 560, 1125, 560, 2250, 560, and so forth, the waveform is shown in:

**Figure 21. Simulation Waveform**



**Table 5.**

ADDRESS	Name	R/W	Description
0x0800	IR Command 1 Low Byte	R/W	The First Command means IR LED on time, the unit depends on System Clock frequency.
0x0801	IR Command 1 High Byte	R	
0x0802	IR Command 2 Low Byte	R/W	The Second Command means IR LED off time, the unit depends on System Clock frequency.
0x0803	IR Command 2 High Byte	R/W	
•	•	R	•
•	•	R	•

ADDRESS	Name	R/W	Description
0x0FFE	IR Command 4096 Low Byte	R/W	IR LED off time, the unit depends on System Clock frequency.
0x0FFF	IR Command 4096 High Byte	R/W	

## Transmitting an IR Frame

To transmit an IR Frame:

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.

## Enabling Self-Learning

To enable self-learning:

1. Write Command Register Bit1 to Low.
2. Write Command Register Bit1 to High
3. Check Status Register Bit0. High means a IR frame has been 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.

## Functional Description

The design consists of the following components:

- I2C/SPI Interface, Control FSM
- Ir Decoder
- Ir Encoder
- Ir Data Buffers

The following sections explain the functional details and interface requirements of these modules.

## Pin Table

The targeted device is the iCE40 Ultra iCE5LP-SWG36. The pin definitions are defined below.

**Table 6. iCE40 Pinout for Target System**

Ball Number	Bank	Pin Name	Pin Description
F4	0	i_sys_clk	System clock input, 27MHz
B2	0	i_sys_rst	Active low system reset input
F5	0	i_ir_din	IR signal from IR receiver module
A2	0	o_ir_dout	IR signal out from IR tx module
B1	0	o_ir_power	Shut down IR circuit.
E2	2	i_scl	I2C SCL signal
C1	2	io_sda	I2C SDA signal
E1	2	i_spiclk	SPI CLK signal



Ball Number	Bank	Pin Name	Pin Description
F1	2	i_spics	SPI CSn signal
F2	2	o_spimiso	SPI MISO signal
D1	2	i_spimosi	SPI MOSI signal

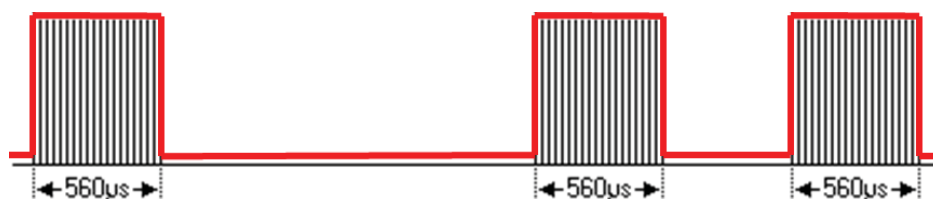
## IR Frame Capture

### Carrier Filter

To implement waveform copy process, the carrier is not required. iCE40 Ultra auto filters the carriers from i\_ir\_din pin. For example, get red waveform from original waveform.

And for transmission, iCE40 Ultra auto pulls carriers.

**Figure 22. Carrier Filter**

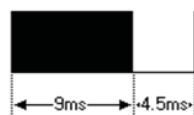


### Preamble Code Detection

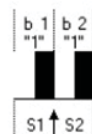
Different IR remote protocol include different preamble code, so for self-learning mode, a High Level contains more than four pulses means frame start. And preamble code also need to record.

**Figure 23. Preamble Code Detection**

**NEC Preamble**



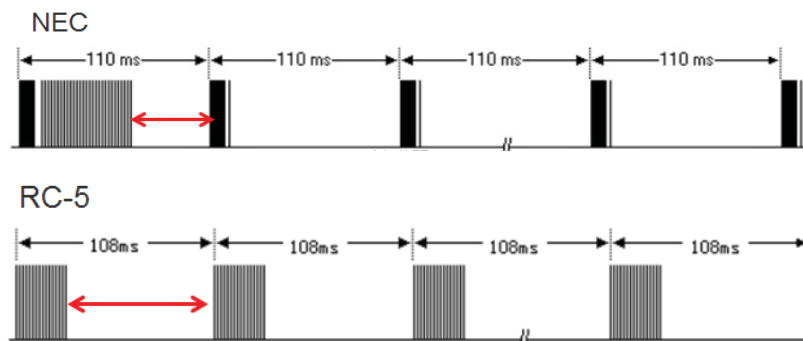
**RC-5 Preamble**



### Stop Code Detection

A very long low level mean Stop Code, then whole frame is recorded, overflow time is 100 ms.

**Figure 24. Stop Code Detection**



## Clock and Reset

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

---

## Resource Utilization

*Table 7. Resource Utilization*

LUTs	PLBs	BRAMs	I/Os	I2C	SPI
877	197	8	7	0	1
891	228	8	5	1	0

## Board Information

For more information on procuring the iCE40 Ultra Mobile Development Platform, please contact your local Lattice Sales Representatives.

For more information on Snapdragon Board APQ8074, please go to [www.intrinsyc.com](http://www.intrinsyc.com).

## References

- DS1048, [iCE40 Ultra Family Data Sheet](#)

## Technical Support Assistance

e-mail: [techsupport@latticesemi.com](mailto:techsupport@latticesemi.com)

Internet: [www.latticesemi.com](http://www.latticesemi.com)

## Revision History

Date	Version	Change Summary
June 2014	1.0	Initial release.

© 2014 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at [www.latticesemi.com/legal](http://www.latticesemi.com/legal). All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.