iCE40 Ultra™ RGB LED Controller

User's Guide

June 2014
UG75_1.0

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.

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 Board properly and become familiar with its main features.
- Work and become familiar with the software required for RGB LED Controller demonstrations.
- Utilize the additional hardware required to run the demonstrations.
- Understand the design details of the RGB LED Controller demo implemented on iCE40 Ultra.
- Run the demo along with the Intrinsyc® 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 RGB LED Controller demonstration in detail and the second section describes the RGB LED Controller design. The RGB LED Controller demonstration is performed using I2C or SPI interface with the Processor.

RGB LED Controller Demonstration

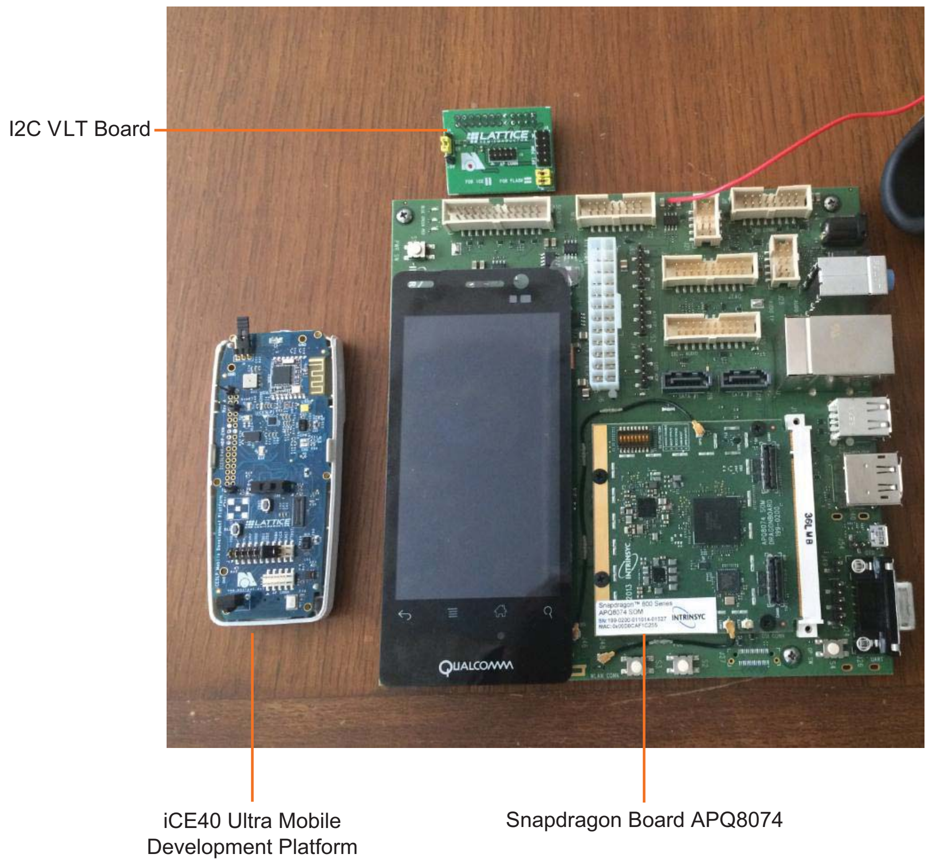
This section describes the RGB LED Controller demonstration in detail.

RGB LED Controller Demonstration Setup Using I2C Interface

The RGB LED Controller demonstration setup using I2C interface consists of the following components, as shown in Figure 1.

- APQ8074 Intrinsyc DragonBoard Gen 2 with +5V adaptor and USB debugger cable
- iCE40 Ultra Mobile Development Platform with programming USB cable
- I2C VLT board
- Three flexible connecting cables

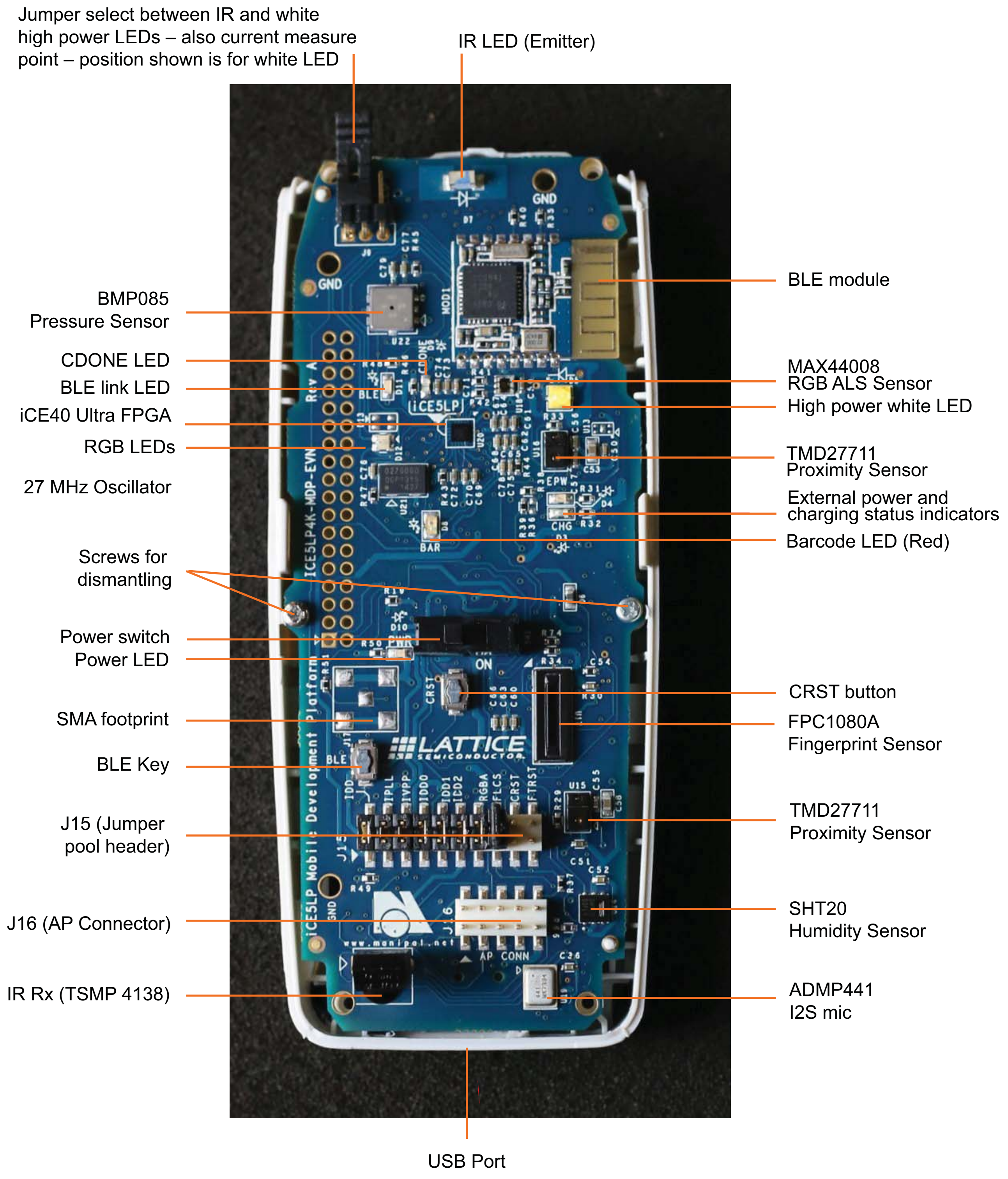
Figure 1. RGB LED Controller I2C Demo Setup



iCE40 Ultra Mobile Development Platform Default Jumper Settings

The details of the iCE40 Ultra Mobile Development Platform default jumper settings are shown in Figure 2

Figure 2. Default Jumper Settings



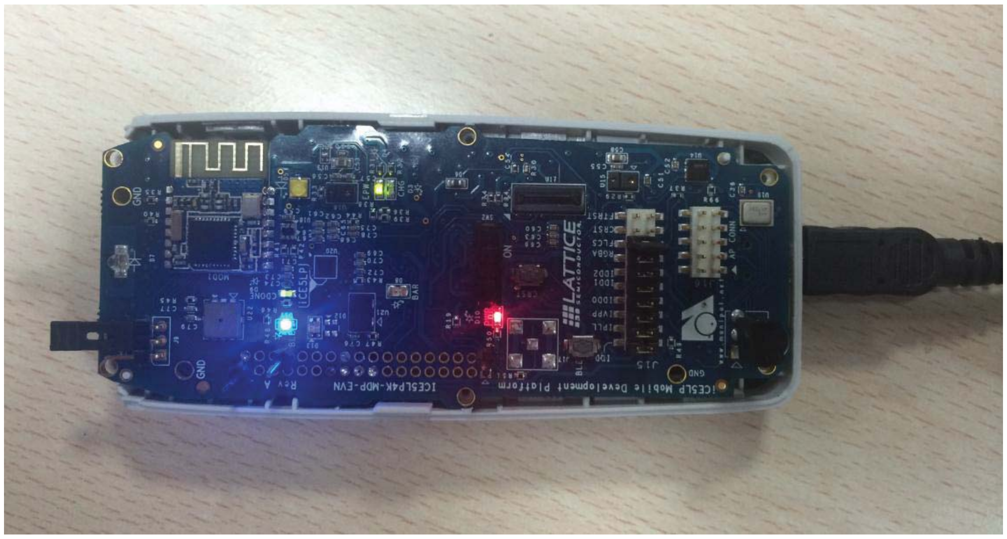
Note: In the J15 jumper set the FLCS pins.

Programming the iCE40 Ultra Mobile Development Platform

To program the iCE40 Ultra Mobile Development Platform:

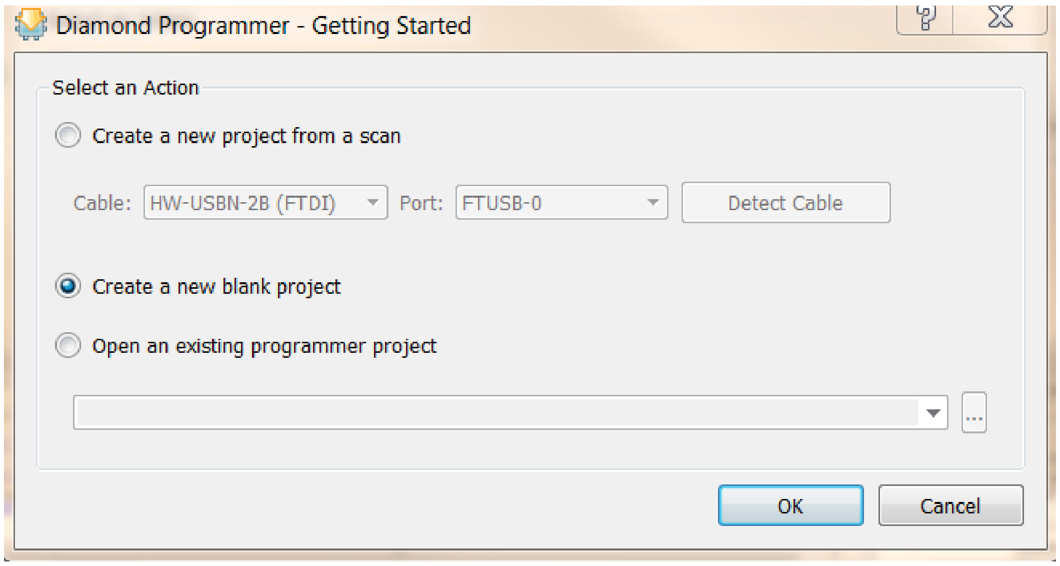
1. Connect the iCE40 Ultra Mobile Development Platform to USB port of PC.

Figure 3. 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 4. Creating New Project



Diamond Programmer - Getting Started

Select an Action

( ) Create a new project from a scan

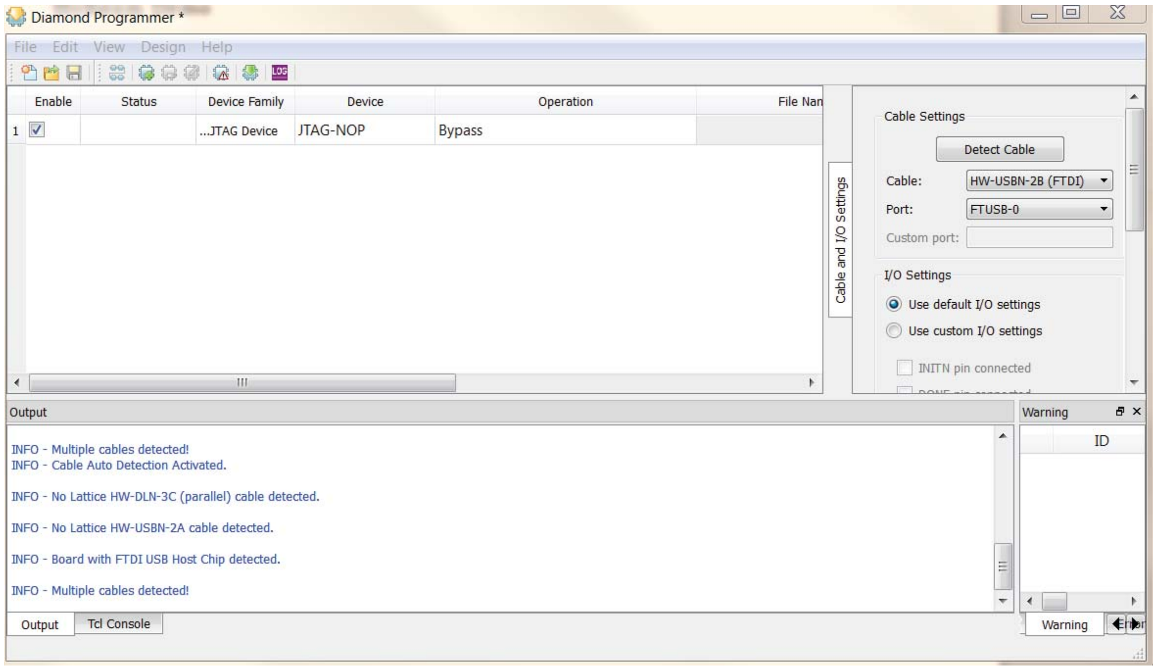
Cable: HW-USBN-2B (FTDI) Port: FTUSB-0

(●) Create a new blank project

( ) Open an existing programmer project

This opens the Diamond Programmer main interface.

Figure 5. Diamond Programmer Interface



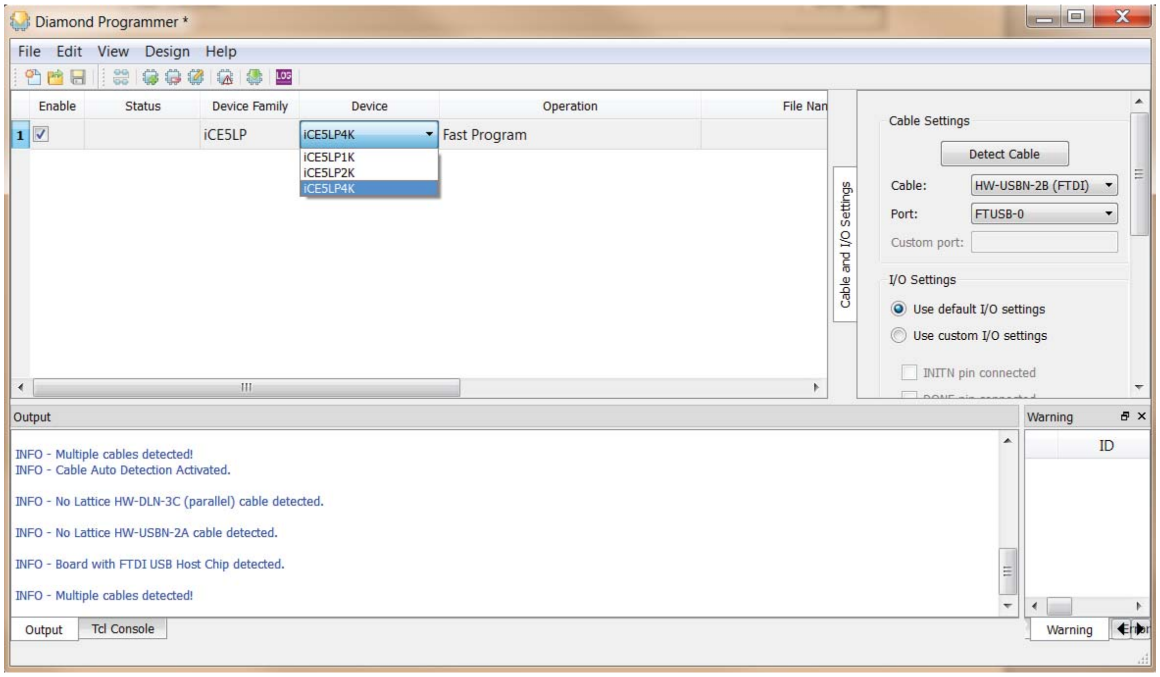
	Enable	Status	Device Family	Device	Operation	File Nan
1	✓		...JTAG Device	JTAG-NOP	Bypass	

Cable: HW-USBN-2B (FTDI)
Port: FTUSB-0
[✓ Use default I/O settings]

```
INFO - Multiple cables detected!
INFO - Cable Auto Detection Activated.

INFO - No Lattice HW-DLN-3C (parallel) cable detected.

INFO - No Lattice HW-USBN-2A cable detected.

INFO - Board with FTDI USB Host Chip detected.

INFO - Multiple cables detected!
```

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

Figure 6. Selecting the Device



	Enable	Status	Device Family	Device	Operation	File Nan
1	✓		iCE5LP	iCE5LP4K	Fast Program	

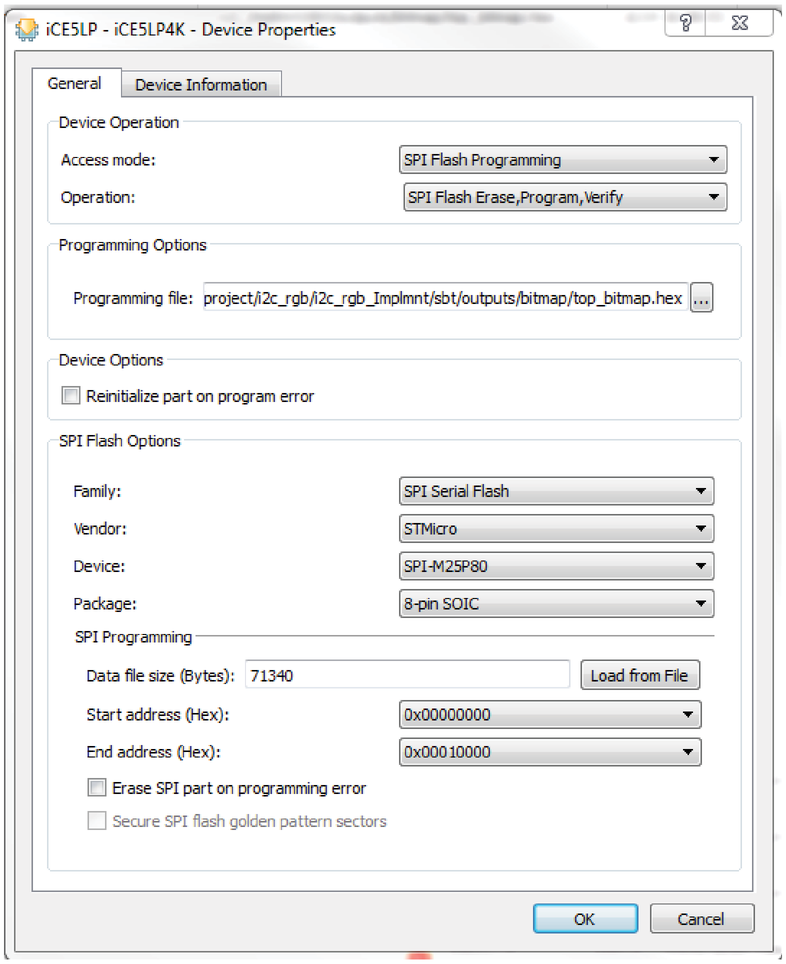
iCE5LP1K
iCE5LP2K
iCE5LP4K

Cable: HW-USBN-2B (FTDI)
Port: FTUSB-0
[✓ Use default I/O settings]

```
INFO - Multiple cables detected!
INFO - Cable Auto Detection Activated.

INFO - No Lattice HW-DLN-3C (parallel) cable detected.

INFO - No Lattice HW-USBN-2A cable detected.

INFO - Board with FTDI USB Host Chip detected.

INFO - Multiple cables detected!
```

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

Figure 7. Selecting Programming File



iCE5LP - iCE5LP4K - Device Properties

General | Device Information

Device Operation

Access mode: SPI Flash Programming

Operation: SPI Flash Erase,Program,Verify

Programming Options

Programming file: project/i2c_rgb/i2c_rgb_Implmnt/sbt/outputs/bitmap/top_bitmap.hex

Device Options

☐ Reinitialize part on program error

SPI Flash Options

Family: SPI Serial Flash

Vendor: STMicro

Device: SPI-M25P80

Package: 8-pin SOIC

SPI Programming

Data file size (Bytes): 71340 | Load from File

Start address (Hex): 0x00000000

End address (Hex): 0x00010000

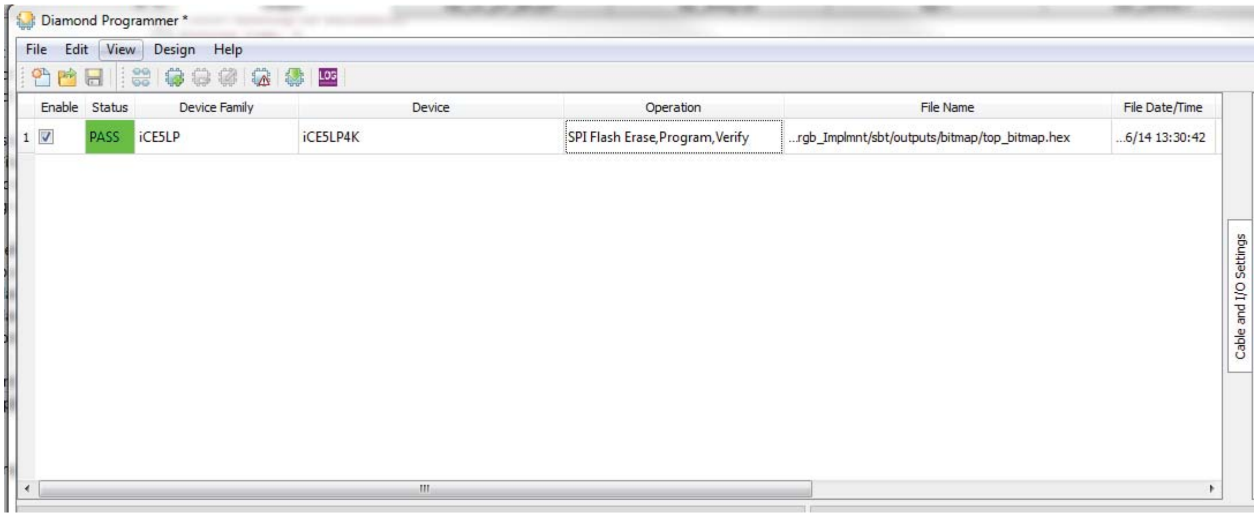
☐ Erase SPI part on programming error

☐ Secure SPI flash golden pattern sectors

OK | Cancel

6. Select the program to use in programming the device.
7. Verify that the operation has successfully completed.

Figure 8. Verifying Operation



Diamond Programmer *

File Edit View Design Help

	Enable	Status	Device Family	Device	Operation	File Name	File Date/Time
1	✓	PASS	iCE5LP	iCE5LP4K	SPI Flash Erase,Program,Verify	..rgb_Implmnt/sbt/outputs/bitmap/top_bitmap.hex	...6/14 13:30:42

Connecting iCE40 Ultra Mobile Development Platform to SnapDragon Board APQ8074

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

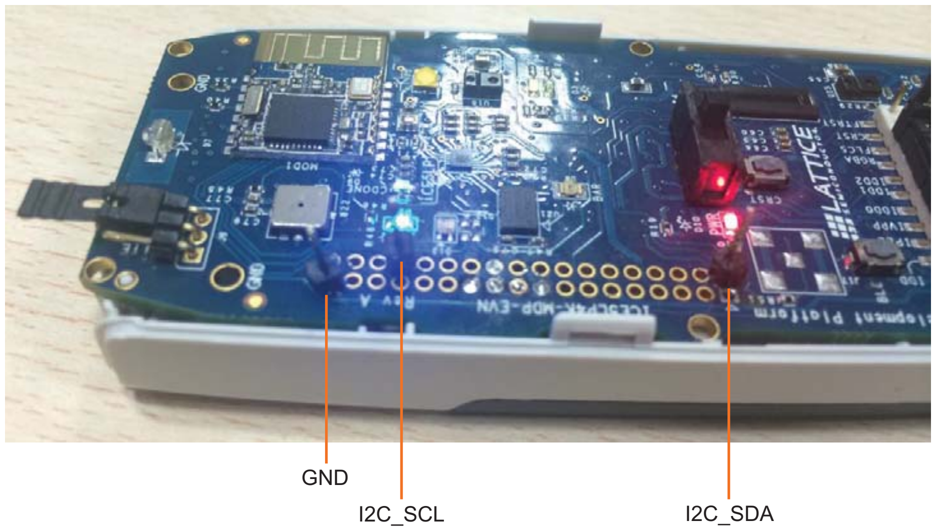
1. Connect the I2C lines between I2C VLT board and iCE40 Ultra Mobile Development Platform as shown in Table 1.

Table 1. I2C Line Connections

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

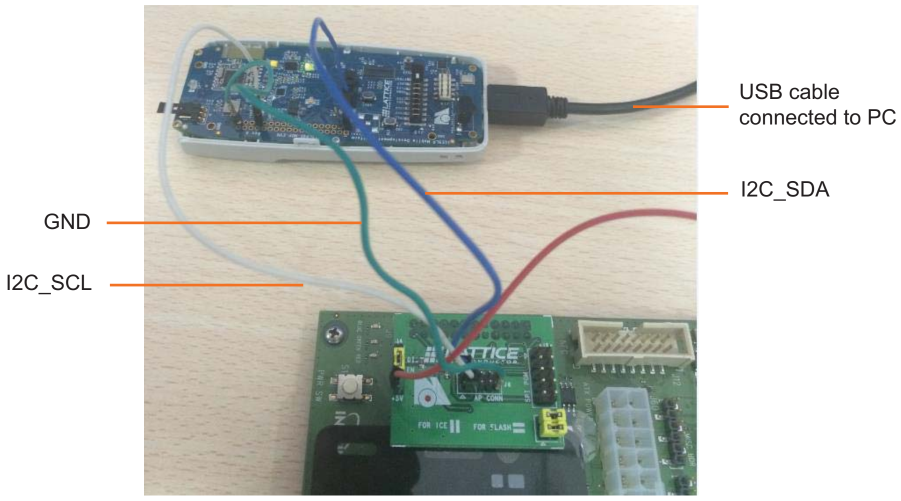
2. Solder three jumper pins as shown in the Figure 9.

Figure 9. Soldering Jumper Pins



3. Connect the I2C lines as shown in Figure 10.

Figure 10. Connecting I2C Lines



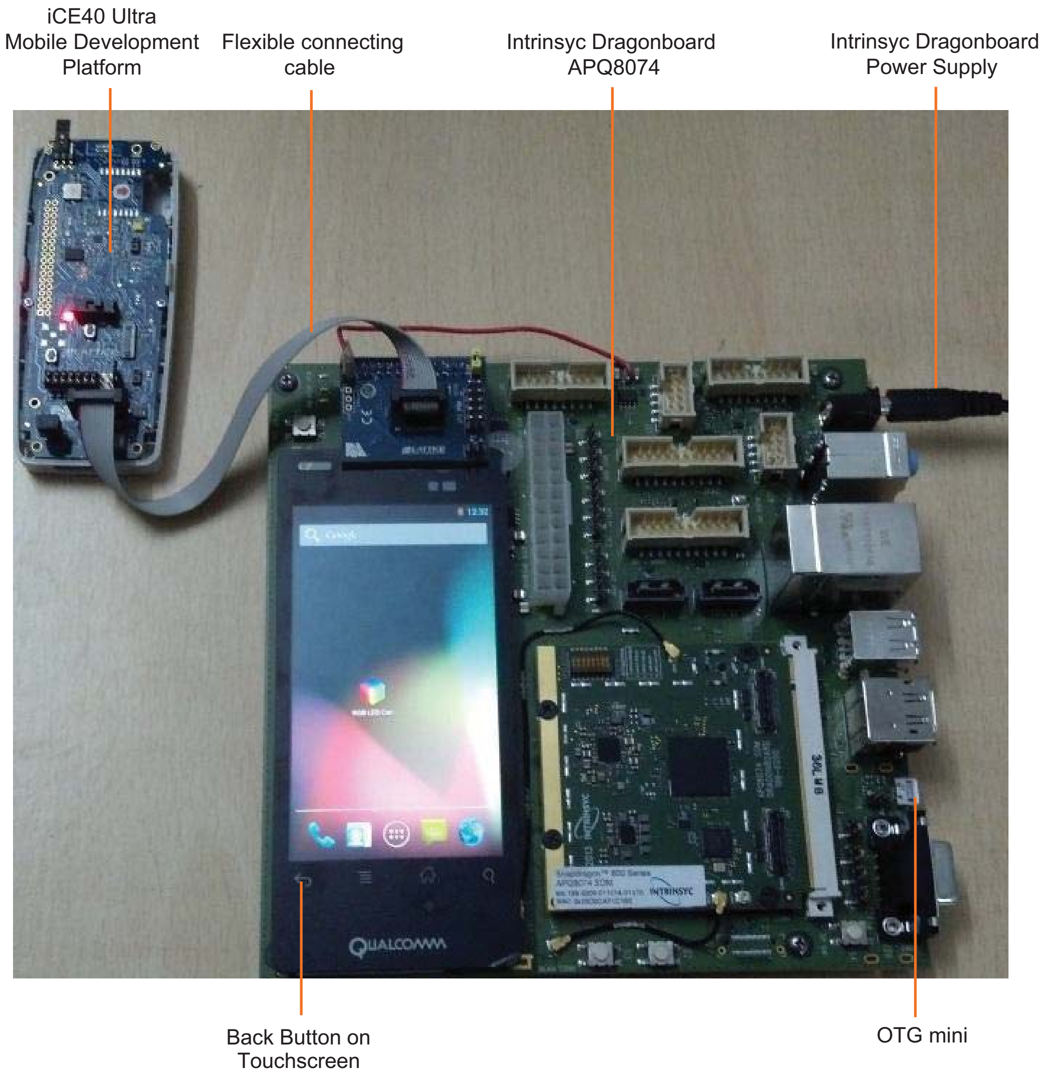
RGB LED Controller Demonstration Setup Using SPI Interface

The RGB LED Controller demonstration setup using SPI interface consists of the following components:

- APQ8074 Intrinsyc DragonBoard
- iCE40 Ultra Mobile Development Platform
- Flexible connecting cable

Connect the iCE40 Ultra Mobile Development Platform to the adapter board mounted on the Intrinsyc APQ8074 DragonBoard as shown in Figure 11. The sensor header is directly above the display.

Figure 11. Connecting the iCE40 Ultra Mobile Development Platform


iCE40 Ultra Mobile Development Platform
Flexible connecting cable
Intrinsyc Dragonboard APQ8074
Intrinsyc Dragonboard Power Supply
Back Button on Touchscreen
OTG mini

iCE40 Ultra Mobile Development Platform Default Jumper Settings

Set Key A to select Bar LED. In the jumper pool J15, set all jumpers except FLCS and CRST.

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

Connecting the iCE40 Ultra Mobile Development Platform to the Intrinsyc DragonBoard

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

1. Power-off the Intrinsyc DragonBoard.
2. Connect one end of the flexible connecting cable to the adapter board mounted on the Intrinsyc 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 Intrinsyc DragonBoard.

The connections are shown in Figure 12 and Figure 13.

Figure 12. Cable Connection to Adapter Board

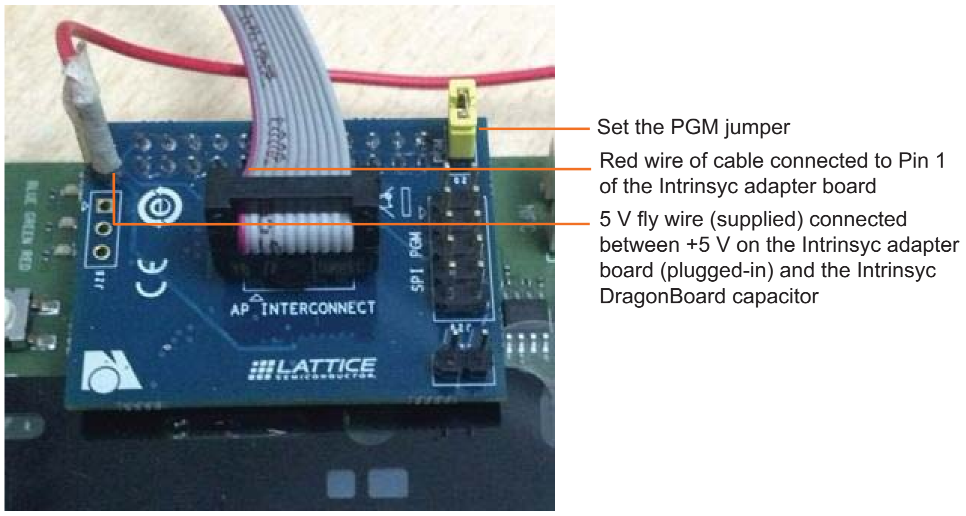
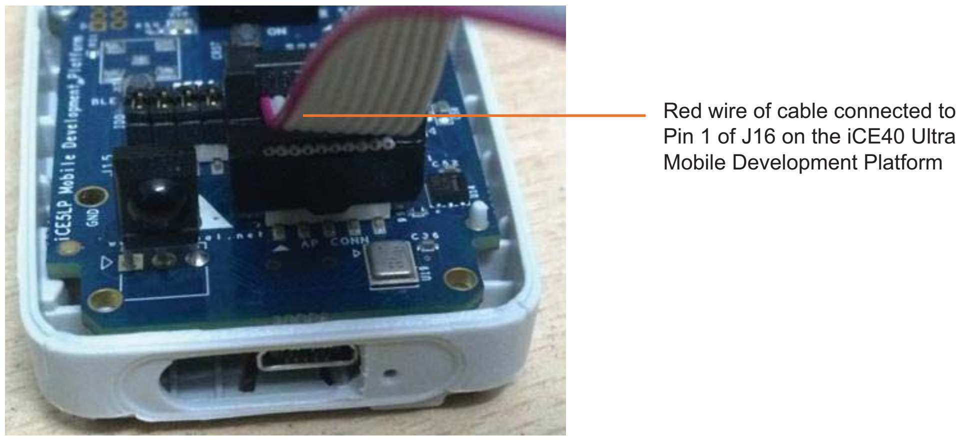

Set the PGM jumper
Red wire of cable connected to Pin 1 of the Intrinsyc adapter board
5 V fly wire (supplied) connected between +5 V on the Intrinsyc adapter board (plugged-in) and the Intrinsyc DragonBoard capacitor

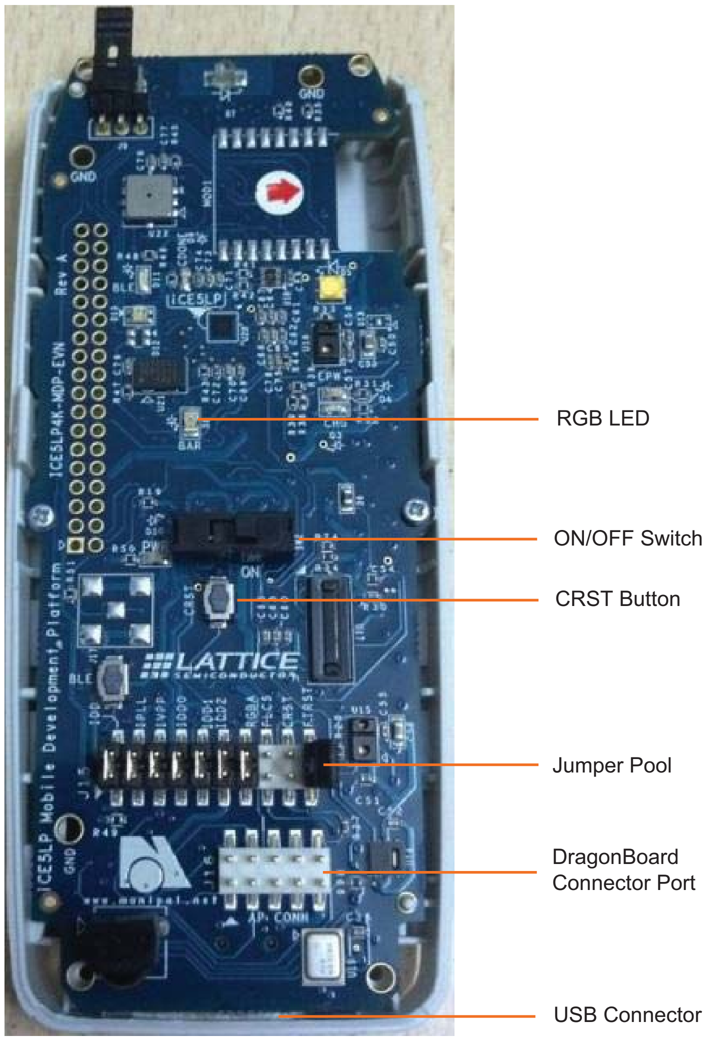
Figure 13. Cable Connection to iCE40 Ultra Mobile Development Platform


Red wire of cable connected to Pin 1 of J16 on the iCE40 Ultra Mobile Development Platform

iCE40 Ultra Mobile Development Platform Details

The details of the iCE40 Ultra Mobile Development Platform are shown in Figure 14.

Figure 14. iCE40 Ultra Mobile Development Platform Details



RGB LED Controller Software Setup

This section provides the procedures in downloading Flash system and boot images to Intrinsyc 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 Intrinsyc DragonBoard:

1. Make OTG (mini-USB) connection from Intrinsyc 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 Intrinsyc DragonBoard in FASTBOOT mode.

   Keep the S2 button pressed on the Intrinsyc 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
   ```

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

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

   ```
   #fastboot devices
   ```

   When the procedure is completed, the board is ready to be flashed with the system and boot images.
5. Download *APQ8074_JB_BootImage.zip* and extract the contents below.

   For I2C: */APQ8074_JB_BootImage/I2C_img*
   For SPI: */ APQ8074_JB_BootImage/SPI_img*

6. 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 SPI:

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

   ForI2C:

   ```
   #cd APQ8074_JB_BootImage /I2C_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.

7. Reboot the board to verify the current board images using the command below:

   ```
   #fastboot reboot
   ```

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

Installing RGB_LED_Controller.apk to Android

To install RGB_LED_Controller.apk to Android:

1. Make OTG (mini-USB) connection from Intrinsyc 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 *RGB_LED_Controller.apk* to Intrinsyc Dragonboard, run the command below.

   For SPI:

   ```
   #cd RGB/RGB_Demo/demonstration/Demo_Apk/spi/RGB_LED_Controller.apk
   #adb root
   #adb remount
   #adb install RGB_LED_Controller.apk
   ```

   For I2C:

   ```
   #cd RGB/RGB_Demo/demonstration/Demo_Apk/i2c/RGB_LED_Controller.apk
   #adb root #adb remount
   #adb root #adb remount
   #adb install RGB_LED_Controller.apk
   ```

5. 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 RGB/RGB_Demo/demonstration/bitstream/spi
   #adb push rgb_bitmap.bin /etc/firmware
   ```

   For I2C:

   ```
   #cd RGB/RGB_Demo/demonstration/bitstream/i2c/top_bitmap.bin
   ```

Demo Procedure

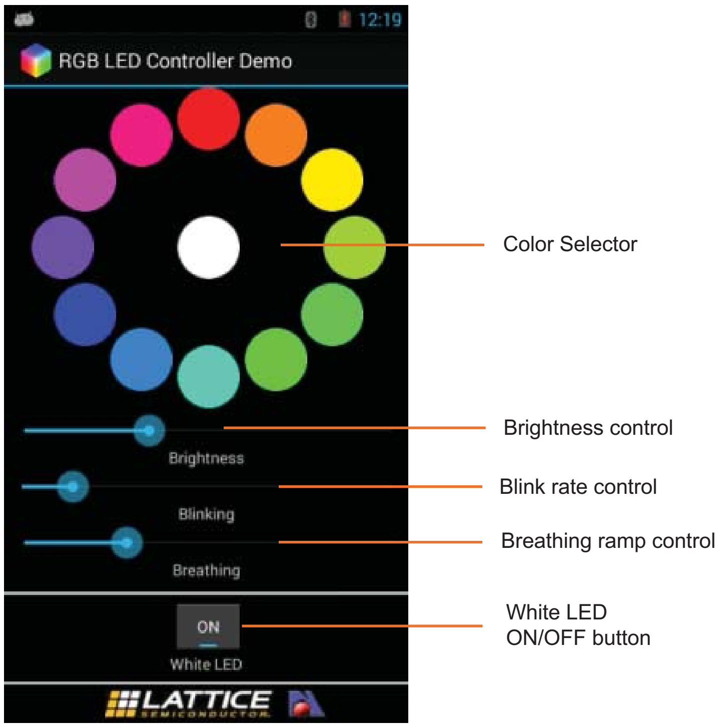
To run the demo:

1. Connect the iCE40 Ultra Mobile Development Platform to DragonBoard using the flexible connecting cable.
2. Power ON Dragonboard and wait for the boot sequence to complete and the Home screen to appear.
3. Unlock the screen. Go to the Android application menu and click the **RGB LED Controller Demo V1** application.
4. Wait for the Processor Configuration to be completed. This is indicated by the glowing of the CDONE LED on the iCE40 Ultra Board.
5. The application is now ready to control the RGB and White LED. 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. White LED can be turned ON/OFF by pressing the **White LED** button. For details, refer to the RGB LED Controller SPI Demo Application Features section.

RGB LED Controller SPI Demo Application Features

The RGB LED Controller SPI Demo application is shown in Figure 15.

Figure 15. RGB LED Controller SPI Demo Application



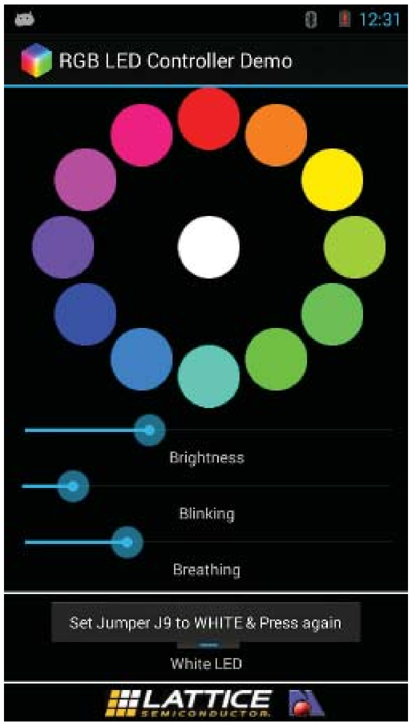
- The Color Selector allows you to choose from 13 different colors in the pallet.
- The Brightness control slider allows you to control the brightness of the LEDs. Adjusting the slider to the left decreases brightness and adjusting the slider to the right increases brightness.
- The Blink rate control 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, which 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.
- The White LED button allows you to turn ON/OFF the white LED on the board.

Before turning ON the white LED, make sure the white LED is selected by setting jumper J9 on iCE40 Ultra Mobile Development Platform to "WHITE". This is indicated by a message prompt at the bottom of the interface as shown in Figure 15.

Note: This step prevents burning of the IR LED due to high current.

Figure 16. Prompt to Set J9 to WHITE



Set the jumper J9 to "WHITE" if it is not currently set and wait for the message to disappear. Press the **White LED** button to turn ON the White LED. Click the button again to turn OFF the LED.

Caution: Looking directly into the LED may cause eye strain.

Troubleshooting

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

1. Close the Barcode Emulator process running in background.
2. In the Android menu, select **System settings** > **Applications** > **Manage Applications** > **Barcode Test** > **Force Stop**.
3. Restart Intrinsyc DragonBoard.
4. Open the RGB LED Controller Demo application from the Android menu. The application is ready to control the RGB and White LEDs on the iCE40 Ultra board.

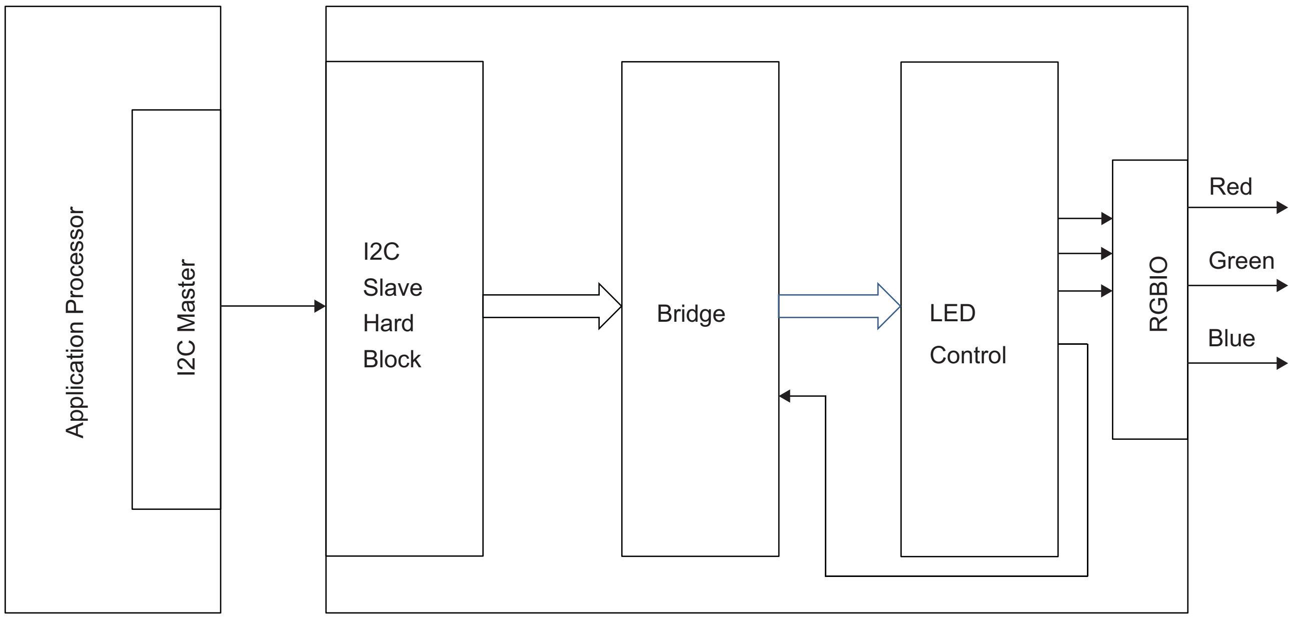
RGB LED Controller Design

The following section describes the internal details of the RGB LED Controller design.

Overview

The iCE40 Ultra Mobile Development Platform provides functionality to drive multi-color LED (R.G.B), 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 I2C slave interface, data control, buffer and LED control to drive LED. The iCE40 Ultra device can be interfaced with the application processor.

Figure 17. Block Diagram



Features

- I2C Slave RX only interface to minimize IOs
- LED control logic provides necessary PWM function
- RGB driver IOs with constant current sink for driving LEDs
- LED control logic with registers programmed over I2C
- Individual LED brightness control for 256 levels (16 million colors total)
- On/Off timing (blink rate)
- Breath control

Figure 18. Functional Block Diagram

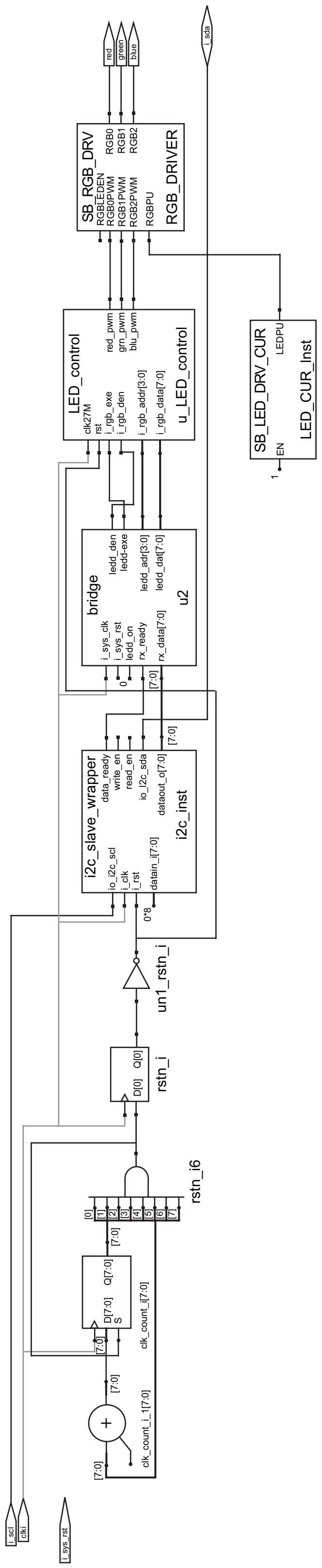


Table 2. Signal Description

PORT	WIDTH	DIRECTION	DESCRIPTION	Source/Destination
i_sys_clk	1	Input	System clock (optional)	Clock from the board
i_sda	1	inout	I2C bus serial data	From I2C Master
i_sck	1	inout	I2C bus serial clock	From I2C Master
red	1	Output	PWM output from RGB Driver	To Red of RGB LED
green	1	Output	PWM output from RGB Driver	To Green of RGB LED
blue	1	Output	PWM output from RGB Driver	To Blue of RGB LED
i_sys_rst	1	Input	Reset input from board (optional)	Determined by user

Functional Description

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

Top level (top)

The RGB LED Driver top level module contains the I2C slave to application processor, the bridge module, and LED control logic. It also contains the RGB driver with constant current sink. The RGB LED driver design also features a power on system reset.

Configurable parameter: `define use_HSSG

This uses the internal HFOSC for clocking the design and is set to 48 MHz. Commenting this compiler directive includes a clki port in the design for a clock 27 MHz coming from the board.

I2C Slave Wrapper

This module instantiates the i2c_Slave module in which the SB_I2C hard IP primitive is instantiated. The I2 C _Slave module implements the state machine to interface with the system bus on the SB_I2C primitive. The state machine performs the initial configuration of the primitive at power on or after a reset event. It also performs the register read/write from the system bus to read out the data received on the I2C controller and pass it on to higher level module validated by a rx_ready signal. The rx_data is 8 bits.

Bridge

This module reads data available on the i2c_slave_wrapper. When rx_ready goes high, the rx_data is converted to 16 bits and written into FIFO. Whenever there is data in the FIFO, the state machine reads data from the FIFO and maps it to the LEDD IP in the following manner:

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

led_dat[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 I2C.

RGB_DRV

This primitive instance if for the IO block that provides a constant sink current open drain driver. For signal description, please refer to the RGB_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.

RBG_CUR_DRV

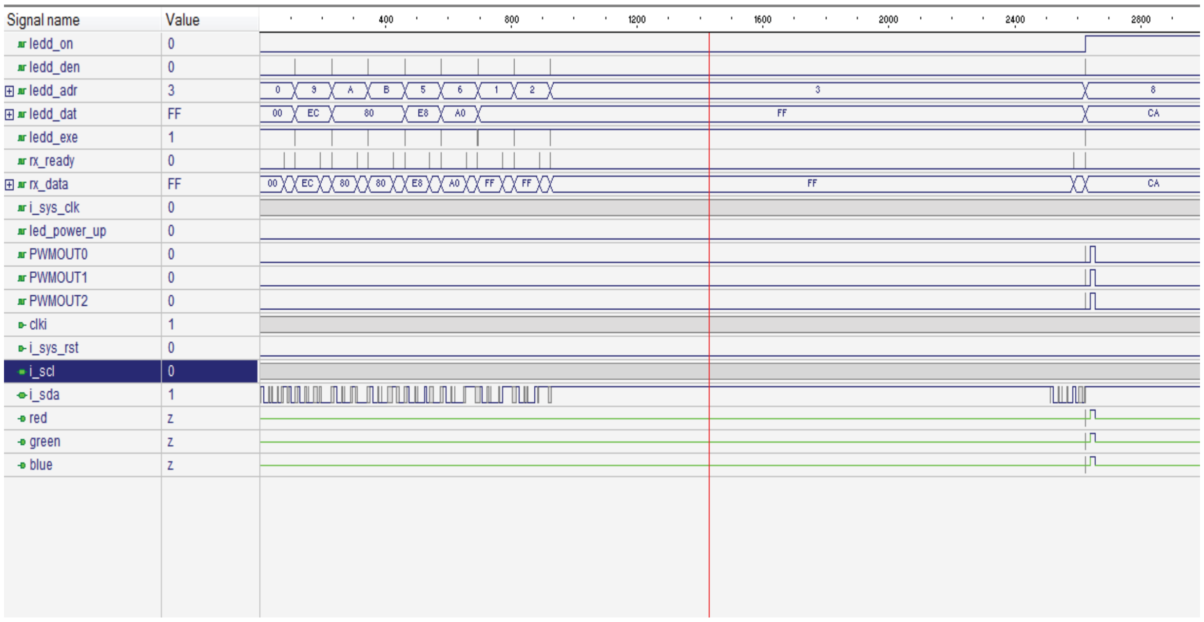
This primitive is required to power up the RGB_DRV.

Operating Sequence:

1. The application processor writes two bytes of data with bit [3:0] of first byte containing 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 slave_i2c_wrapper, converts it to 16 bits and writes it into the FIFO.
3. When the FIFO goes low, the bridge drives ledd_exe low and waits for the ledd_on to go low.
4. When ledd_on goes low, the bridge state machine writes the commands buffered in the FIFO in the LED control registers.
5. When the 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.

Simulation Waveform

Figure 19. RGB LED Driver Simulation Waveform



Resource Utilization

Table 3. Resource Utilization

LUTs	PLBs	BRAMs	I/Os	I2C	SPI
687	152	1	7	1	0
705	137	1	10	0	1

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.

References

- DS1048, iCE40 Ultra Family Data Sheet

Technical Support Assistance

e-mail: techsupport@latticesemi.com

Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
June 2014	1.0	Initial release.