Using GreenPAK's I2C Port to Control an RGB LED's Color

Authors: Chuck Husted and Luke Thomas

Date: March 01, 2016

Introduction

This app note explains how to reconfigure the SLG46531's registers via I2C. Specifically, it shows how to change the registers associated with counters that set the PWM values for driving the R, G, and B cathodes of an LED.

It is important to note that any reconfigurations via I2C are volatile and will revert to the programmed code after POR inside the GreenPAK resets.

GreenPAK Benefits

GreenPAK is a very versatile, low current consumption IC. It can offload functions from other microcontrollers and larger SOC's. For example, a microcontroller can be active (drawing several mA of current), and can write via I2C to the SLG46531V to set the RGB, then the microcontroller can go into deep sleep mode to save system current. The GreenPAK would manage the RGB light function and wake up the microcontroller as needed. The GreenPAK can also act as an interrupt by using another I/O pin.

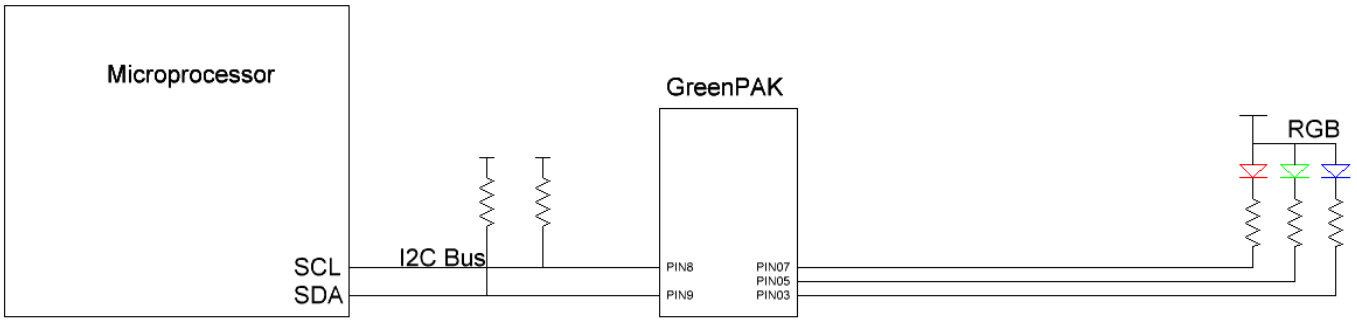

Figure 1. GreenPAK Used as an RGB Driver in a Larger System

GreenPAK Configuration

The GreenPAK design shown in Figure 2(a) implements a simple RGB LED driver. On the GreenPAK Universal Dev Board, connect Pin 3 to the cathode of the blue LED, Pin 5 to the cathode of the green LED and Pin 7 to the cathode of the red LED. The common anode should be connected to the VDD pin of the GreenPAK.

Remember to use current limiting resistors in series with the LEDs as needed. Counter0 was set to a 100 Hz refresh rate so that flicker is not visible to the naked eye. Counter2 has a register that can be set from 0 to 255 which represent 255 PWM steps. Each step represents 0.39%. CNT2, 3, and 4 control the PWM for the RGB LED.

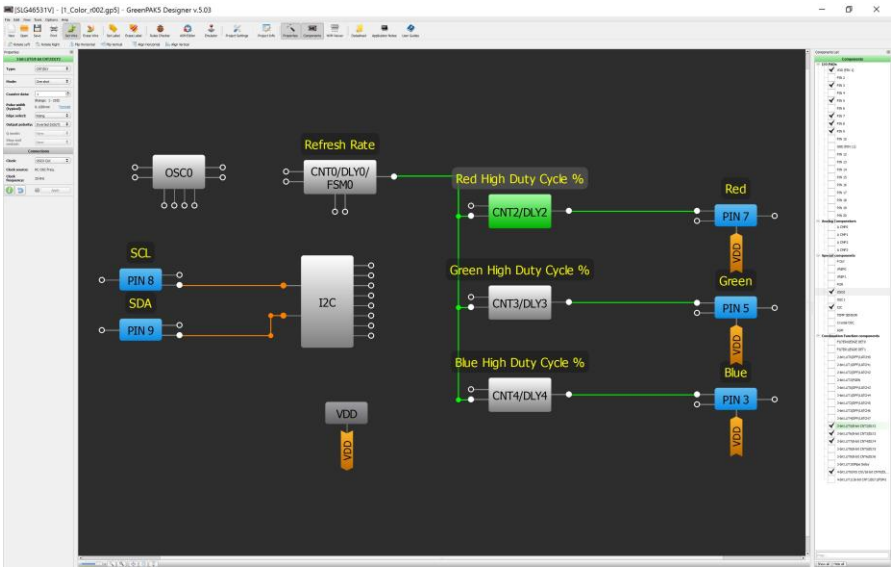

Figure 2(a). GreenPAK block diagram with counters for setting the PWM of each color

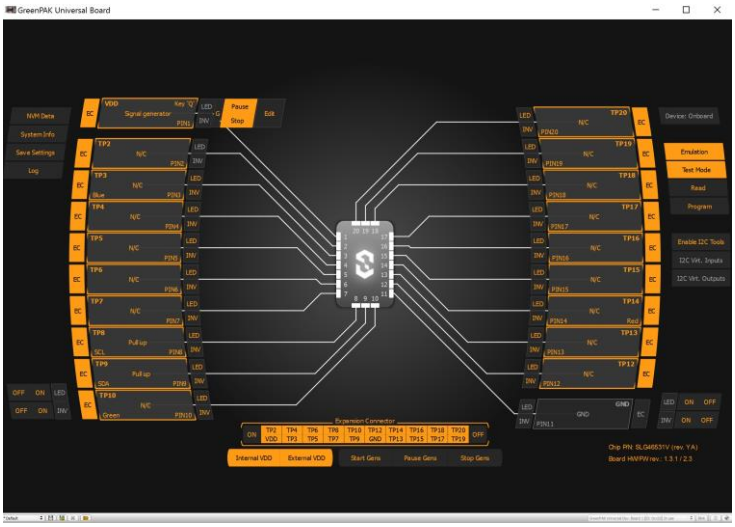

Figure 2(b). GreenPAK emulator window

Creating an I2C Command to Write Register Bits

This application note will not cover the basic GPAK I2C command format. Instead, we will emphasize the specific commands to implement the RGB LED. In the example GreenPAK design file we have used a default chip address of 00 (decimal) which we will show in hex format as 0x00. Note [represents an I2C start bit and] represents a stop bit.

Since the design uses four counters we need to know the address of each counter's register.

Counter	Byte Address (Hex)
CNT0	0xC5
CNT2	0xC0
CNT3	0xC1
CNT4	0xC2

Table 1. Counter register address

Table 1 provides the register address for each of the four counters used in this design. Changing these registers will impact the brightness of the three LEDs and the final color.

Since counters 2 through 4 range from 0 to 255 (decimal), we can write to them via I2C with data from 0x00 to 0xFF.

By putting together all of this information, we can create a command to turn off the red LED by writing the following to the GreenPAK to turn off the red LED:

[0x00 0xC0 0x00]

Similarly, this would turn off the green LED:

[0x00 0xC1 0x00]

And this command would turn off the blue LED:

[0x00 0xC2 0x00]

To turn on all three LEDs to full brightness (and a PWM signal of 100% for each) we can simplify the write command since the three register addresses are sequential:

[0x00 0xC0 0xFF 0xFF 0xFF]

This command writes a 100% PWM signal to the counter associated with the red LED first, then 100% to the counter for the green LED, and lastly 100% to the counter for the blue LED.

Using the GreenPAK Development Kit Emulator

We can use the GreenPAK Development kits emulator to create the signals that we need to drive the LEDs properly. First we must connect the development kit to a computer or laptop's USB port. Second, place a SLG46531V IC into the socket. Third, open the design file inside the GreenPAK Designer software and click the Emulator button in the upper toolbar. This should appear as in Figure 2(b). Next we must enable the I2C tools within the software. This is the button to the right side of the screen. Please see Figure 2(c).

After enabling the I2C tools, we must select the I2C Virtual Inputs button. This tool allows us to write directly to the counter registers. Please see Figure 3(a).

Enter the value of 256 (decimal) into the "New Value" box for counter 0 (CNT0). This sets the refresh timing for the PWM's. Next enter 255 into the "New Value" box for Counter 2 (CNT2) and press the "Write" button.

This will turn on the red LED with a 100% PWM setting.

See Figure 3(b) for the window settings and Figure 3(c) for a picture of the development kit connected to an RGB LED.

Next enter 0 into the "New Value" box for CNT2 and enter 255 into the "New Value" box for CNT3. Press "Write" and you will see the green LED turn on at 100% PWM.

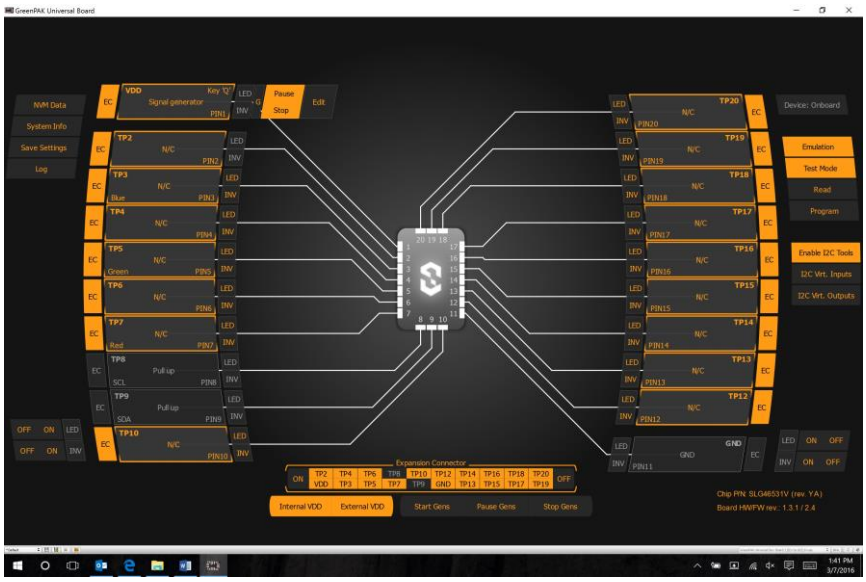

Figure 2(c). GreenPAK emulator window in Emulation Mode

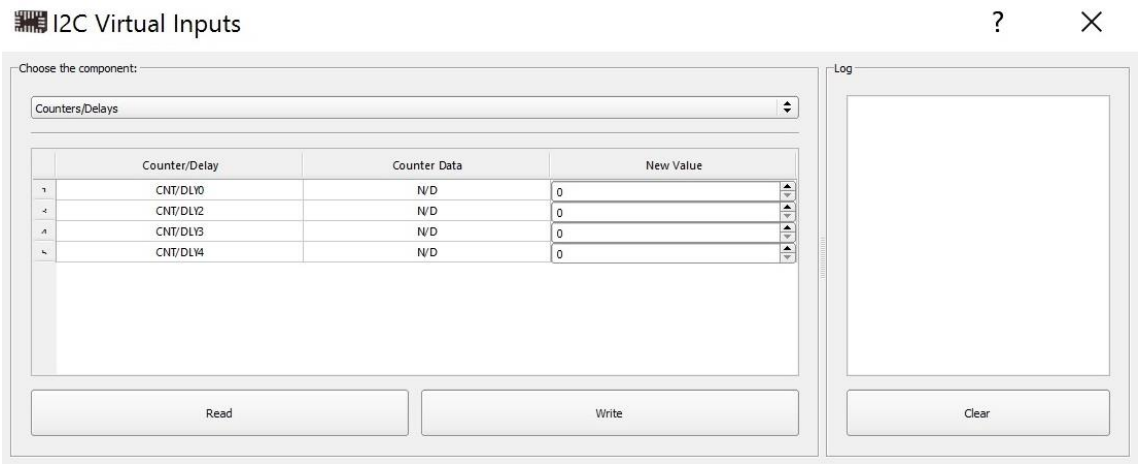

Figure 3(a). GreenPAK emulator with I2C Virtual Inputs window

Please see Figure 3(d) for the window settings for the green LED and Figure 3(e) for a picture of the green LED on.

Likewise for the blue LED at 100% PWM, please see Figure 3(f) for the window settings and Figure 3(g) for a picture.

To simplify the write process for changing the PWM setting for all three colors at the same time, you simply use a sequential write command.

For example, to write 50% PWM to all three LEDs this would be the command:

[0x00 0xC0 0x80 0x80 0x80]

This command starts by writing 0x80 to the register for the red LED PWM, immediately followed by writing 0x80 to the register for the green LED PWM and finally writing 0x80 to the register for the blue LED PWM.

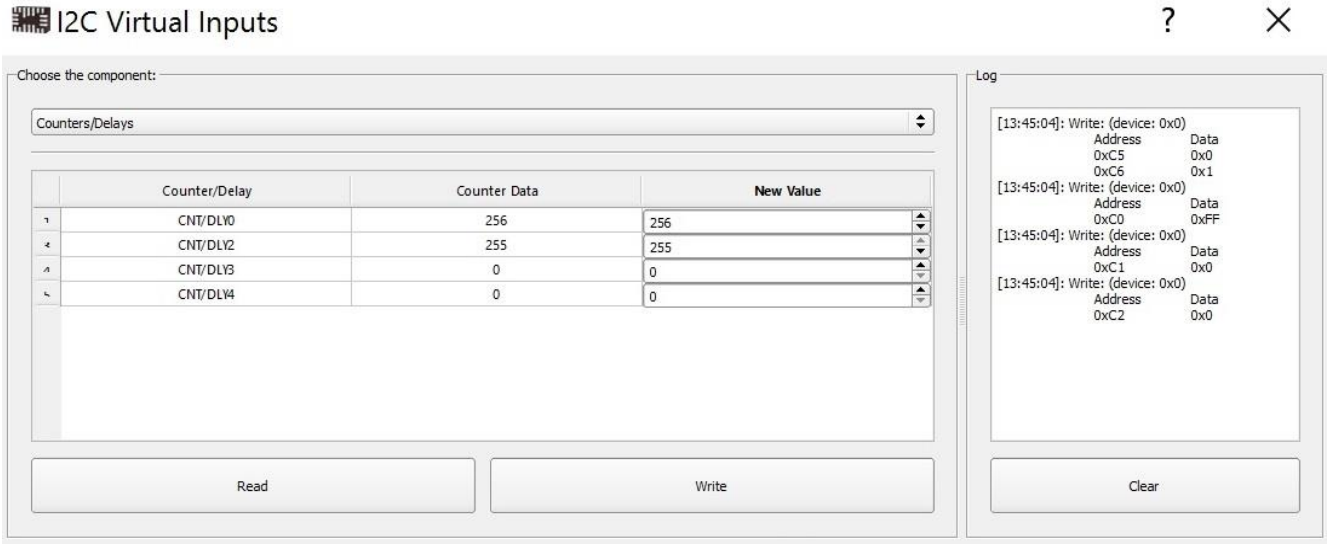

Figure 3(b). GreenPAK I2C Virtual Inputs for 100% PWM setting for red LED

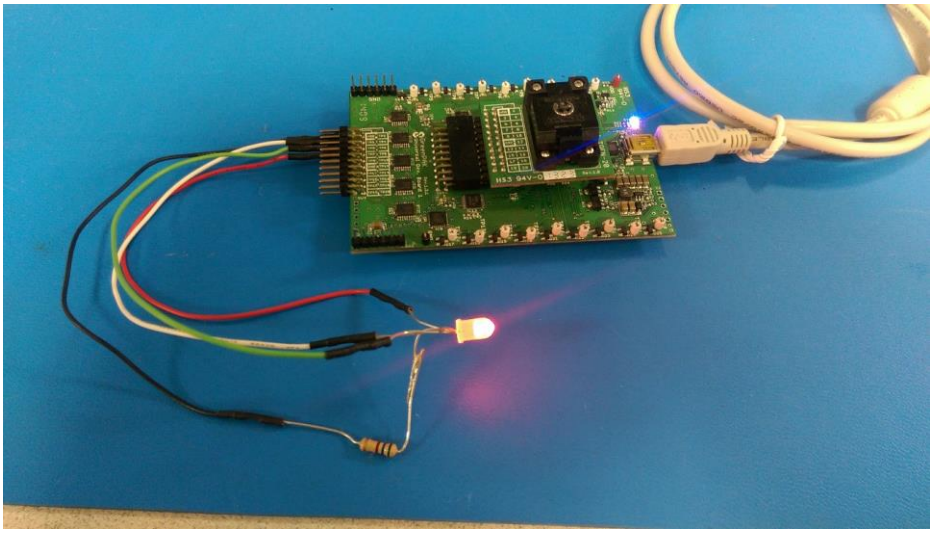

Figure 3(c). GreenPAK Development Kit driving the red LED in an RGB LED

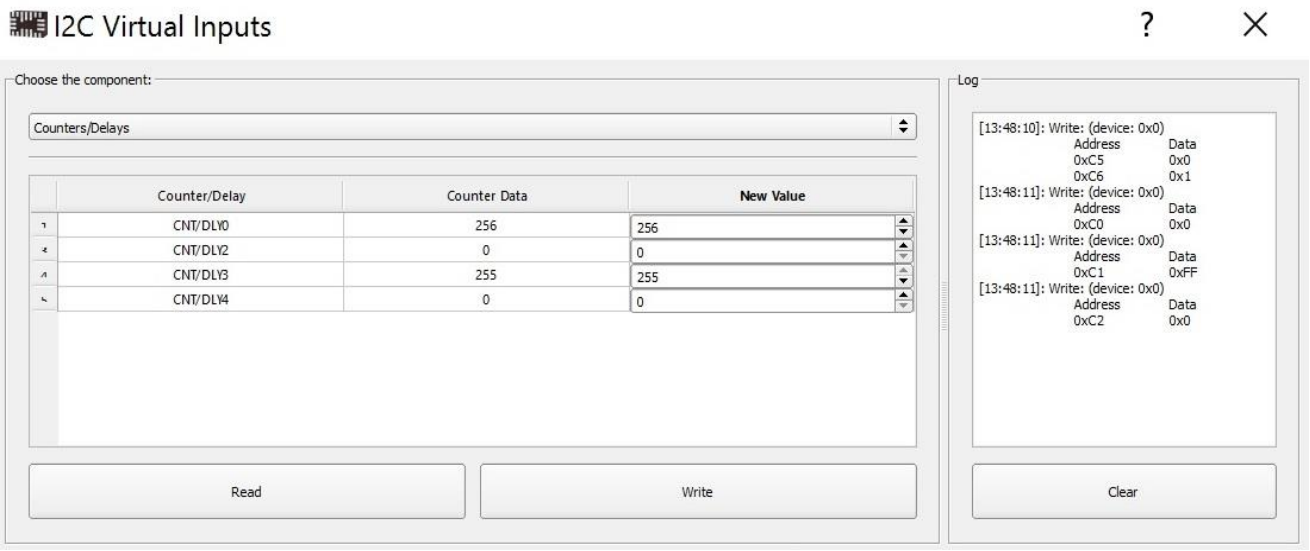

Figure 3(d). GreenPAK I2C Virtual Inputs for 100% PWM setting for green LED

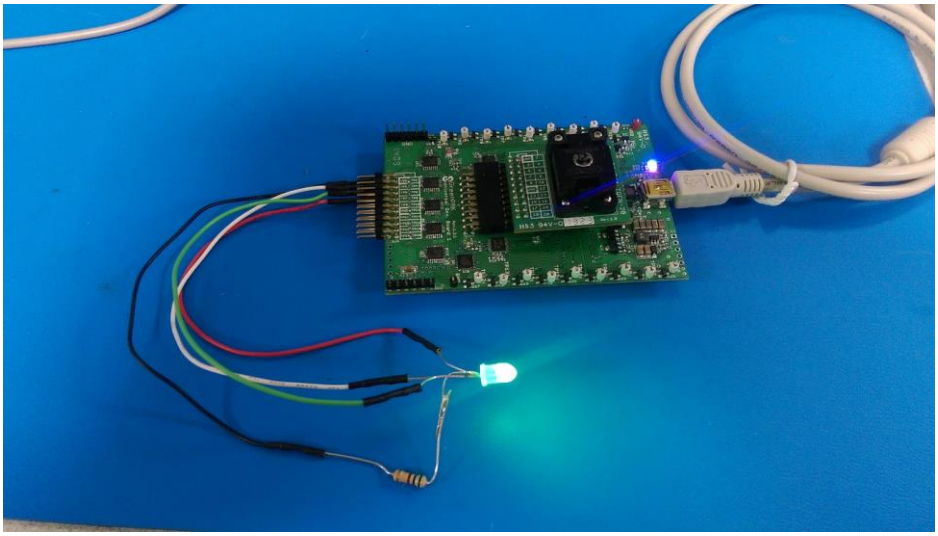

Figure 3(e). GreenPAK Development Kit driving the green LED in an RGB LED

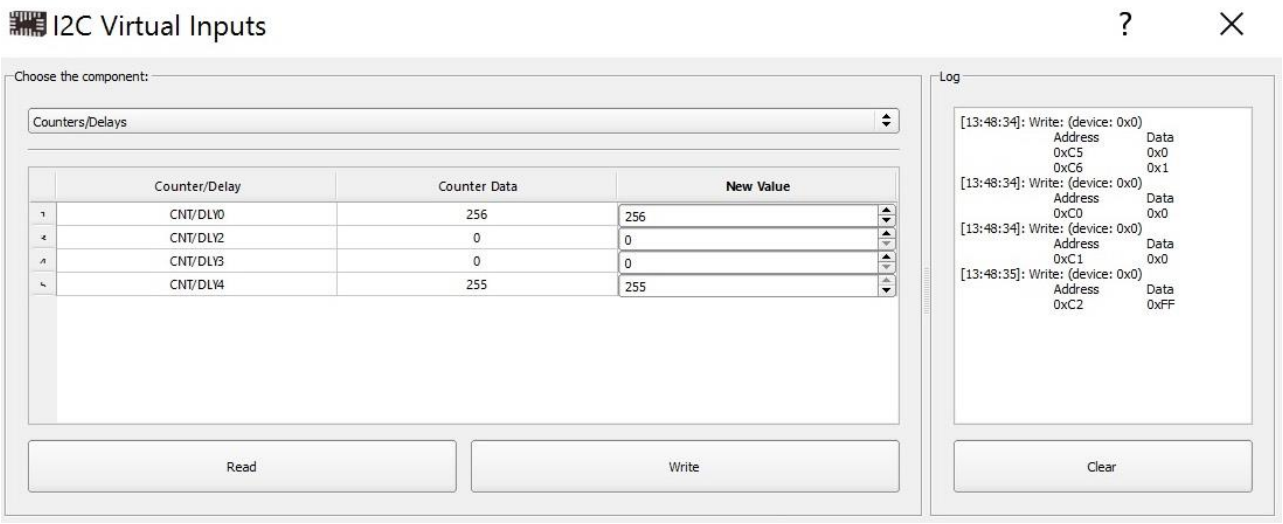

Figure 3(f). GreenPAK I2C Virtual Inputs for 100% PWM setting for blue LED

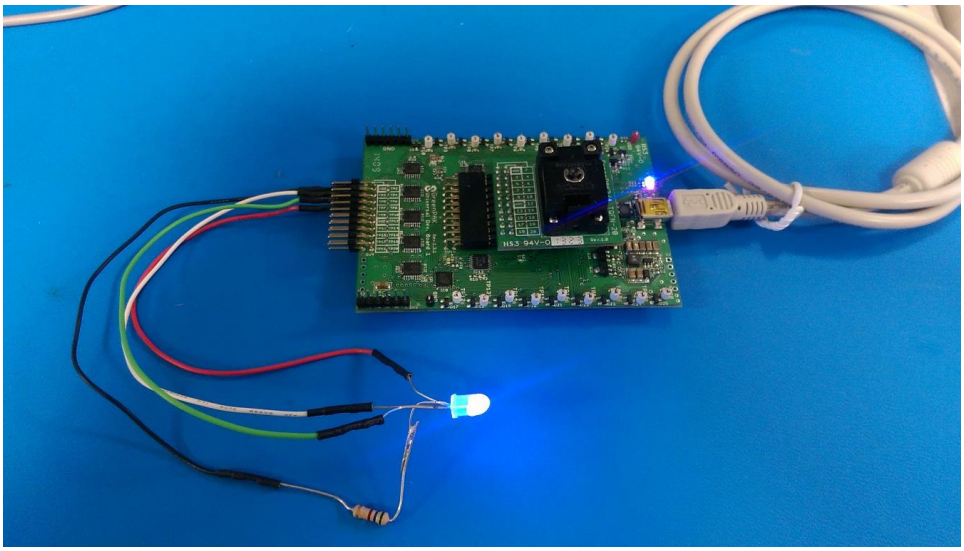

Figure 3(g). GreenPAK Development Kit driving the blue LED in an RGB LED

Conclusion

We can use a GreenPAK SLG46531V to implement RGB LED driver functions. By implementing a system architecture this way, a microcontroller or other system SOC can be put in sleep mode to save overall system power consumption which is desirable for battery based portable and wearable systems.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01 Jan 2024)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.