PrISM™ Technology for LED Dimming

AN47372

Author: Raj Kumar Singh Parihar
Associated Project: Yes
Associated Part Family: CY8CLED04, CY8CLED08, CY8CLED16
GET FREE SAMPLES HERE
Software Version: PSoC Designer™ 4.4 or PSoC Express™ 3.0
Associated Application Notes: AN16035

Application Note Abstract

This application note presents an overview of Precision Illumination Signal Modulation (PrISM™) technology for LED dimming applications. It also discusses the challenges faced in implementing high resolution PrISM and recommends solutions to address these issues.

Introduction

Modulation schemes that keep average duty cycle or signal density proportional to desired dimming level in a fixed time period are used to dim LEDs. These schemes are popularly known as Pulse Density Modulation (PDM). Pulse Width Modulation (PWM), which modulates width of pulses according to desired dimming levels, is the simplest example of PDM.

For LED dimming, the order in which individual pulses occur within a fixed time period is not important. However, the total high time or signal energy in the fixed time frame must be as configured. Cypress' PrISM serves as an alternative to modulation schemes used for LED brightness control. PrISM technology is implemented using high resolution Stochastic Signal Density Modulation (SSDM) modules.

Although PWM has some excellent features, it suffers from significant harmonic generation at a relatively low frequency. Sometimes it may require intense filtering to remove high frequency components. Due to high Electro Magnetic Interference (EMI) generation from PWM waves, SSDM is used to implement PrISM. The idea is to spread the energy at different frequencies so that it is easy to filter the higher harmonics, if required.

Assumptions

This document assumes that the reader is familiar with LED dimming concepts (refer application note AN16035). Prior knowledge and understanding of PSoC® and EZ-Color™ device architecture, C programming language, and switching regulator design is also useful. In this application note, PrISM and SSDM are used interchangeably.

EZ-Color: Overview

Cypress' EZ-Color family of devices offers an ideal control solution for HB LED applications that require intelligent dimming control. EZ-Color devices combine the power and flexibility of PSoC (Programmable System-on-Chip™) with Cypress' PrISM modulation technology to provide lighting designers a fully customizable and integrated lighting solution platform.

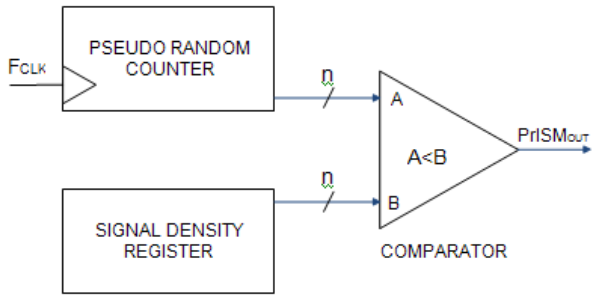
PSoC Express™ software with lighting specific drivers significantly reduces development time and simplifies implementation of fixed color points through temperature and LED binning compensation. EZ-Color's limitless analog and digital customization allows simple integration of features in addition to intelligent lighting, such as battery charging, image stabilization, and motor control during the development process.

For more information about EZ–Color solutions and associated technologies, visit http://www.cypress.com/ez-color/

PrISM Technology

Before addressing the issues in implementing PrISM, it is important to know the basic building blocks. PrISM uses stochastic signal density modulation to generate the average signal density equivalent to dimming value. Figure 1 on page 2 shows the basic components of a typical PrISM system.

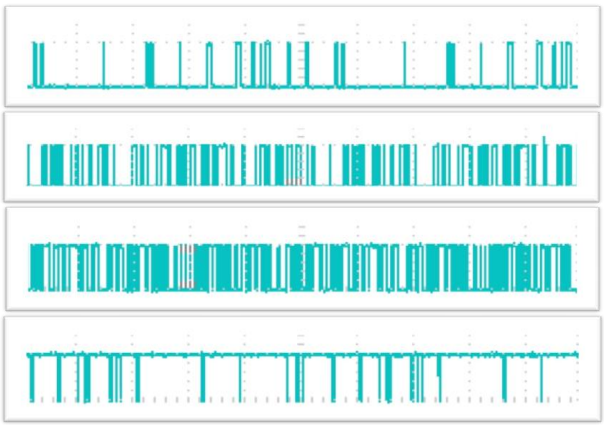
Figure 1. PrISM Block Diagram



The block diagram is divided into three major modules.

- Pseudo Random Counter: The top left block is a pseudo random counter, which generates n-bit pseudo random code at every tick of FCLK. It produces all codes ranging from 0 to 2^n–1. Generated codes are almost random. The counter repeats the codes every 2^{n+1} time.
- Signal Density Register: This is the left bottom block, which is a simple n-bit register. After reset , it is loaded with the desired signal density value and holds the same value until the user writes a new signal density value.
- Comparator: The block on the right is an asynchronous comparator, which continuously compares the content of the random counter with the value of signal density register. It makes the output HIGH when the signal density value is greater than the value of pseudo random counter. Pseudo random sequences for various dimming levels are shown in Figure 2.

Figure 2. Pseudo Random Sequences for Signal Density Values: 10%, 33%, 50%, 95%



Average Output Frequency

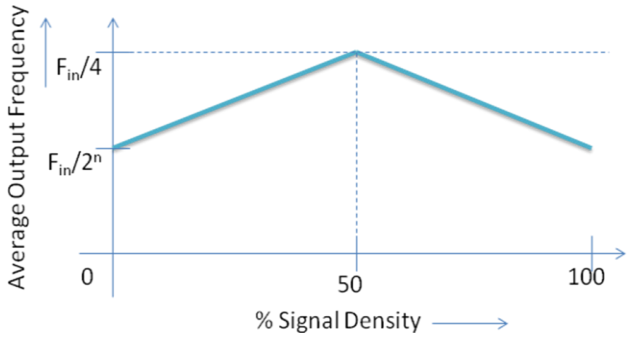
It is important to know the frequency content of SSDM sequence. Unlike PWM, SSDM's output frequency is not fixed; it is variable and is a function of signal density.

SD% = (Signal Density Value / Period) * 100 → Equation 1

F_{OUT} = ½ SD. F_{IN} for (SD < = 0.5)

F_{OUT} = ½ (1- SD). F_{IN} for (SD > 0.5) → Equation 2

Figure 3. Average Output Frequency of SSDM



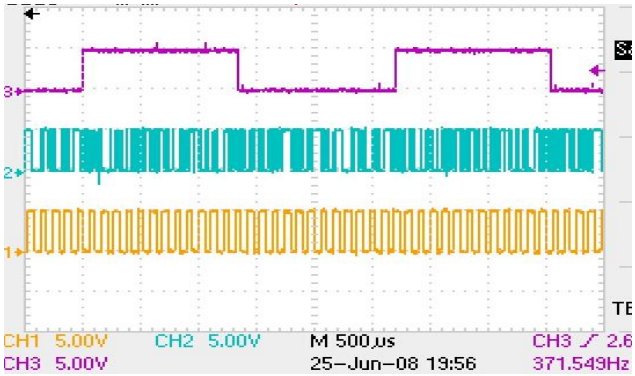
Maximum output frequency components present for any signal density in SSDM sequence are $F_{IN}/2$ and minimum are $F_{IN}/2^n$.

The average output frequency for 50% signal density is average of $F_{IN}/2$ (maximum value) and $F_{IN}/2^n$ (minimum value). For high n, the value of $F_{IN}/2^n$ is low compared to $F_{IN}/2$ and average output frequency is nearly $F_{IN}/4$.

Spectral Plot and Frequency Components

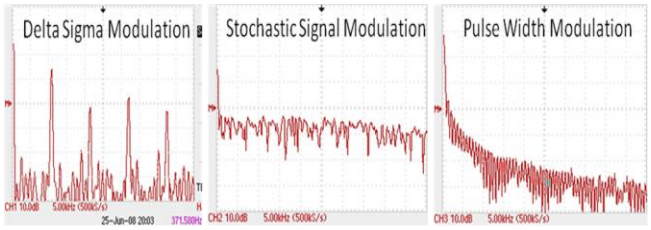
An important advantage of PrISM is less EMI compared to other modulation schemes such as PWM and Delta Sigma Modulation (DSM). In SSDM based PrISM, energy is spread at various frequencies. Therefore, there are no spikes at a particular frequency as they appear in PWM or DSM spectral plot.

Figure 4. PWM, SSDM, and DSM Wave at 50% Duty Cycle



For 50% signal density, on an average the improvement from PWM and DSM is about 35 dBm. Figure 5 shows the comparison of various modulation techniques for 50% duty cycle.

Figure 5. Spectrum Plots of DSM, SSDM, and PWM for 50% Signal Density



Issues

Take care of these issues when any modulation technique is used for LED dimming application. From Figure 2 and Figure 4 on page 2, it is evident that in SSDM the switching is more compared to the equivalent PWM.

- **Minimum Output Frequency:** Minimum output frequency should be at least greater than 120 Hz to avoid the visible flickers. However, output frequency of 300 Hz is considered to be flicker free.
- **Maximum Output Frequency:** Maximum output frequency components (or thinnest pulses in case of spread spectrum) should be at least 10 times lower than switching frequency of regulator circuitry for efficient operation of regulation. Otherwise, these thin pulses are filtered out resulting in significant loss of signal density.

Boundary Conditions for F_{OUT}

In implementing PrISM, take care of these boundary conditions:

F_{OUT} >= 120 Hz

To avoid the visible flickers

- F_{OUT} ~= 300 Hz
  To have flicker free LED dimming
- F_{OUT} <= 1 KHz
  To minimize the inductor noise
- F_{OUT} <= 0.1* F_{SW}
  To make sure that regulator is not filtering out some of the high frequency components

It is impossible to eliminate the audible noise which may be present because of inductors. This is because dimming frequency band overlaps with the human audible frequency band. However, noise is minimized by choosing the right packaging for inductors.

Implementation and Analysis

PSoC Designer™ incorporates SSDM modules and requires API to control them as part of the user module library. They are available in various resolutions such as 8-bit, 16-bit, 24-bit, and 32-bit. In the current implementation, only 8-bit and 16-bit modules are used. 8-bit SSDM module is used to generate the dimming resolution of 2 to 8 bits; 16-bit module generates anywhere from 2 to 16 bits.

Either a PWM module or the clock resources available in PSoC® (VC1, VC2, or VC3) is used to provide the clocks to SSDM modules.

Table 1. User Module Parameters for Sample Project

Parameters	Values
Clock	VC3
SSDMOut	Row_x_Output_y
CompareType	Less Than Or Equal
DimmingResolution	8 or 16
SignalDensity	0
ClockSync	Sync to SysClock

Row_x_Output_y are the interconnect nets used to route the output of SSDM modules to external pins. Both x and y can have values from 0 to 3 in this implementation.

Table 2. Global Resources Setting for Sample Project

Resources	Values
VC3 Source	SysClk/1
VC3 Divider	4

VC3 gets the clock from SysClk and the divider is 4. If SysClk is set to 24 MHz, the SSDM modules tick at the rate of 6 MHz.

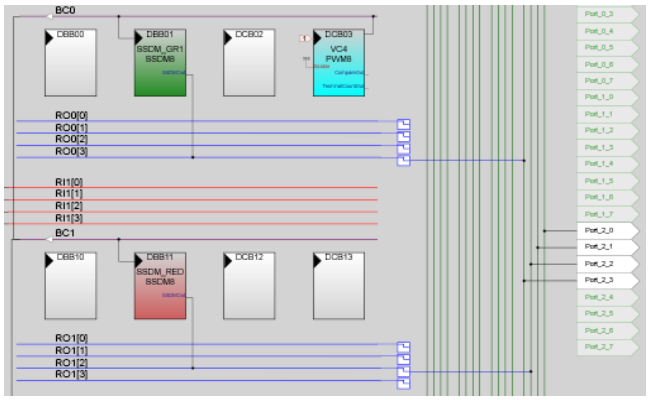
Some of the most frequently used APIs are as follows. The function of these APIs is evident from their names.

- SSDM_Start
- SSDM_Stop
- SSDM_WriteSignalDensity
- SSDM_WriteResolution

8-Bit PrISM

Figure 6 on page 4 depicts implementation of 8-bit PrISM hardware. The two SSDM modules, clocking PWM, inter connection nets, and output ports are visible. In the current implementation, SSDM modules are clocked from a single PWM module. PWM based clock is generated appropriately by setting the duty cycle and time period of PWM module. Using an n-bit PWM, output clock ranging from $F_{IN}/2$ to $F_{IN}/2^n$ is generated. In 8-bit SSDM, codes generated by pseudo random counter for one signal density value ranges from 0 to 255.

Figure 6. 8-Bit PrISM Hardware Implementation



The main concern in LED dimming is that output frequency of SSDM should be at least 300 Hz to avoid flickers.

F_{IN} = $F_{OUT\ (MIN)}$ * 2^n = 300 * 256 ~= 77 KHz

F_{IN} of 77 KHz can be generated easily within the PSoC using VC clocks.

For F_{IN} = 77 KHz from Figure 3 on page 2

$F_{OUT\ (MAX)}$ = F_{IN} / 2 = 77 KHz / 2 = 38.5 KHz

F_{OUT} is below the typical switching frequency of regulator circuitry, which is 1 MHz. This ensures that the regulator is not filtering the thin pulses.

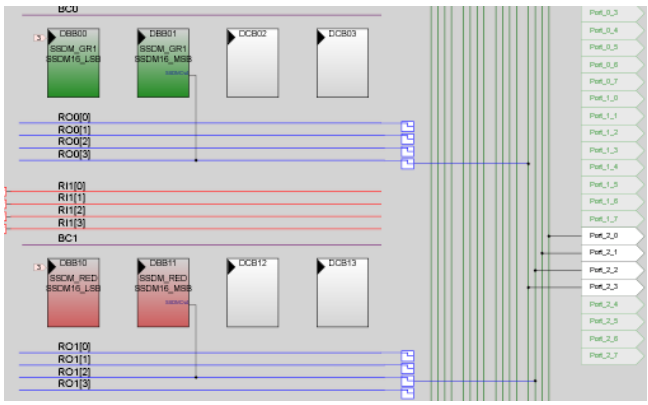
Although anything above the $F_{OUT\ (MIN)}$ of 300 Hz is acceptable, it is beneficial to bring the $F_{OUT\ (MIN)}$ closer to 300 Hz. This is because it brings the $F_{OUT\ (MAX)}$ well below the maximum limit, which is one tenth of the switching frequency of regulator circuitry.

16-Bit PrISM (Multi-Byte PrISM)

PrISM of higher than 8-bit resolution is called multi-byte PrISM. Each SSDM module consumes one digital PSoC block per 8 bits of resolution. So multi-byte PrISM consumes two digital blocks.

Codes generated by 16-bit pseudo random counter range from 0 to 65535. Figure 7 shows the implementation of multi-byte PrISM.

Figure 7. Multi-Byte PrISM Hardware Implementation



1. Start with a guess

   Assume the input frequency (F_{IN}) to multi-byte PrISM to be 1 MHz.

   $F_{OUT\ (MIN)}$ = F_{IN} / 2^n = 1 MHz / 65536 ~= 15 Hz

   This output frequency is too low for LED dimming and results in visible flickers.

   To avoid the visible flickers, increase the $F_{OUT\ (MIN)}$. This is achieved by increasing the input frequency F_{IN}.

2. Increase the input frequency

   New input clock frequency to PrISM block is F_{IN} = 8 MHz

   $F_{OUT\ (MIN)}$ = F_{IN} / 2^n = 8 MHz / 65536 ~= 120 Hz

   This is good enough and hardly results in visible flickers but not better than 300 Hz. Also F_{IN} = 8 MHz is a high value for a system clock of 24 MHz.

   Increasing input clock frequency also increases the maximum frequency components. For F_{IN} of 8 MHz:

   $F_{OUT\ (MAX)}$ = F_{IN} / 2 = 8 MHz / 2 ~= 4 MHz

   This is too high for switching frequency of 2 MHz. The maximum is about 500 KHz which is four times less than the maximum possible switching frequency of 2 MHz.

   From this analysis, it is evident that flickers are avoided due to low frequency components by increasing the input frequency. However, this raises concerns because of high frequency components. For input frequency of 8 MHz, the high frequency components are as high as 4 MHz. This results in loss of average signal density for desired signal density value.

   To avoid filtering of high frequency components, the only solution is to reduce the resolution. This means, without increasing the switching frequency it is not possible to increase the dimming frequency to a very high value.

3. Reduce the resolution

   From the analysis, increasing input frequency is not a good idea in most cases. The option left is to decrease the resolution up to the point where significant signal density is not lost.

   For $F_{OUT\ (MAX)}$ = 500 KHz (for switching frequency of 2 MHz)

   F_{IN} = $F_{OUT\ (MAX)}$ * 2 = 1 MHz

   It is not a good idea to have dimming input clock frequency more than 1 MHz.

   Number of bits which does not lead to visible flickers

   2^n = F_{IN} /F_{OUT} (MIN)

To avoid the visible flickers, minimum output frequency is greater than 120 Hz.

2^n = 1 MHz / 120 Hz ~= 8334 Hz

After taking the log of both sides

0.3010 * n = 3.92

n = 13

From this analysis it is evident that up to 13-bit PrISM can be implemented without significant loss in average signal density for any signal density value.

The number of bits is increased only when switching frequency of regulator circuitry or FET increases.

Improvements

Variable Frequency PrISM

PrISM has more flickers at low and high signal densities. This is because the average output frequency at the extremes is proportional to either signal density or difference of maximum possible signal density and desired signal density (refer to Equation 2). The solution is to increase the input clock frequency at lower signal densities. This is what the variable clock PrISM (vPrISM) does.

The vPrISM topology uses one PWM to clock each PrISM block. The PWM output is routed to the PrISM block via the broadcast bus. Figure 8 shows one such implementation.

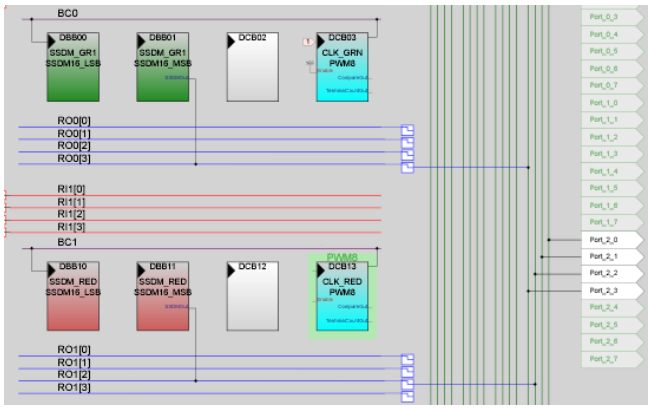
Digital block consumption:

Digital blocks = (2+1)*n

n = number of channels

In normal PrISM, the digital block consumption is 2n+1.

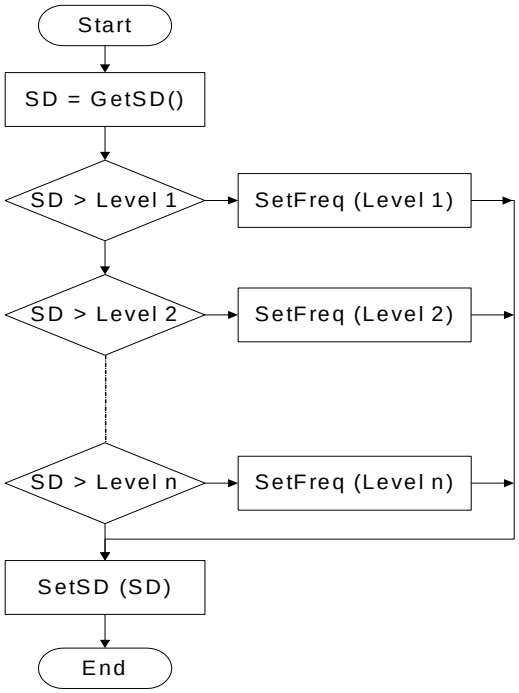
Figure 8. Device View: 16-bit Variable PrISM



vPrISM input clock frequency is determined by following the state machine shown in Figure 9. As the signal density increases, the input clock frequency decreases.

For example, if Level1 is 10% of the maximum resolution of the PrISM, when the SD is greater than Level1, the input clock frequency of the PrISM is decreased.

Figure 9. State Machine to Generate Clocks for vPrISM



Variable Resolution PrISM

Similar to variable frequency PrISM, the resolution of SSDM modules is also changed in run time for some particular signal density values. This is beneficial especially where input frequency is slowed down. It helps to ensure that there are no flickers due to slow input clock. This is because low output frequency components are larger than what they were in the original 16-bit resolution. In summary, slowing down the clock is beneficial for regulator operations and decreasing the resolution is beneficial in keeping the lowest frequency component, which is $F_{in}/2^n$, above 120 Hz.

The idea is combined with the variable frequency approach and resolution can be changed in the same state machine. It is possible to change the resolution of SSDM modules using the following API.

API: SSDM_WriteResolution

Description: Initializes the stochastic counter Polynomial Register with a polynomial that matches the desired resolution.

C Prototype:

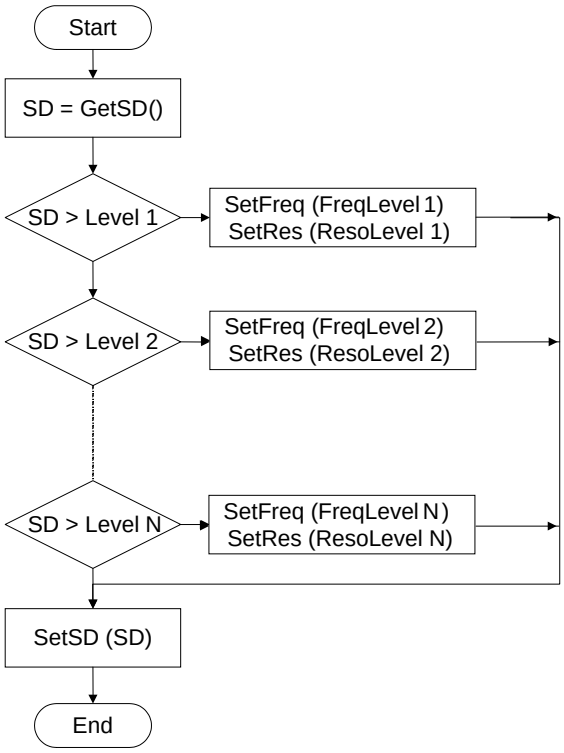
```
Void SSDM_WriteResolution(BYTE bResolution);
```

Parameters:

- **bSignalResolution:** value from 2 to 8, 16, 24, or 32 depending on which version of the user module is used. The resolution value is passed in the A register.
- **Side Effects:** The A and X registers may be modified by this or future implementations of this function. The same is true for all RAM page pointer registers in the Large Memory Model (CY8CLED16).

Figure 10 is the modified version of the state machine shown in Figure 9. This must be called in while loop after dimming values are calculated from color mixing algorithm and before those dimming values are fed into SSDM modules. Separate state machines should be implemented for each channel.

Figure 10. State Machine to Generate the Variable Clock and Variable Resolution



Implementation of vPrISM

In vPrISM, resolutions and operating frequencies of SSDM blocks are varied in run time. Each SSDM module gets clock from a separate clock source—in this case 8-bit PWM modules. Depending on the signal density percentage, the clock frequency and resolutions are adjusted.

In the current implementation, five dimming levels and five frequency levels are implemented. Frequency varies from 12 MHz to 1.2 MHz and resolution from 12-bit to 16-bit.

Table 3. Variable Frequency and Resolution

Signal Density (in %)	F_{IN} (in MHz)	Resolution
0<SD<10 or 90<SD<100	6	16 - bit
10<SD<20 or 80<SD<90	3	15 - bit
20<SD<30 or 70<SD<80	2	14 - bit
30<SD<40 or 60<SD<70	1.5	13 - bit
40<SD<60	1.2	12 - bit

Table 4. Output Frequency of vPrISM at Various Dimming Levels

Signal Density (in %)	F_{OUT} (in Hz)
0<SD<10 or 90<SD<100	92
10<SD<20 or 80<SD<90	92
20<SD<30 or 70<SD<80	122
30<SD<40 or 60<SD<70	183
40<SD<60	292

Firmware

The following versions of PrISM firmware are written.

- Up to 8-bit PrISM using 8-bit SSDM modules
- Up to 16-bit PrISM using 16-bit SSDM modules
- Variable PrISM using 16-bit SSDM modules

The second firmware is used to work with any resolution from 9-bit to 16-bit. When it is used in 16-bit mode, there is loss in average signal density which may not be perceptible.

Guidelines and Recommendations

- A separate clock source such as PWM or timer must be used to clock the SSDM modules if all VC clock sources are already used by some other modules.
- All SSDM modules must be clocked from the same clock source for normal PrISM. vPrISM requires separate clock sources for each SSDM module.
- At 50% signal density, it is easy to avoid filtering of thin pulses. To achieve this, input clock frequency must be decreased. This gives the same duty cycle, but the thin pulses are now wider because of slow input clock, thus pulses are not filtered out from regulator circuitry.
- It is difficult to avoid filtering of thin pulses when the signal density is either very high or very low (1/65535, 65534/65535). Slowing down the clock causes flickers. By reducing the resolution, less loss in average signal density is achieved and the perceivable flickers are avoided.
- Average output frequency of SSDM, $F_{in}/4$, gives an estimation of the kinds of components present in spread spectrum of SSDM. However, take care of $F_{IN}/2$, the highest frequency components and not just F_{IN} /4. Make sure that the switching frequency of the regulator is more than 10 times of F_{IN} /2.

Summary

This application note provides an overview of PrISM technology used for LED dimming applications. It explains the basic functionality of PrISM and its implementation details. Information about 8-bit PrISM, 16-bit PrISM, variable frequency PrISM, and variable resolution PrISM are also described. The application note also discusses the challenges faced in multi-byte PrISM and ways to overcome them. Three PSoC Designer example projects are included for reference.

About the Author

Name: Raj Kumar Singh Parihar

Title: Applications Engineer

Background: Raj Kumar holds a bachelors degree in Electrical and Electronics from BITS – Pilani, India.

Contact: Email: rksp@cypress.com

Ph: +91 9940242401

Document History

Document Title: PrISM™ Technology for LED Dimming

Document Number: 001-47372

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	2546747	RKSP	08/01/08	New application note.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone: 408-943-2600
Fax: 408-943-4730
http://www.cypress.com/