PWM Dithering for High-Color Resolution & Flicker-Free LED Light

LED lighting and LCD backlights should appear flicker-free in order to provide a pleasing look for both human observers and video cameras. When LEDs look stripy on phone cameras or "flutter" to the eye, it's usually because the LED controller is using a slow PWM clock.

Traditionally, achieving very fine brightness control or high color accuracy with 16-bit PWM requires running a slow PWM clock which increases the risk of visible flicker to the eye and rolling-shutter banding on video. This newsletter introduces PWM dithering technology which keeps LED brightness and color smooth and pushes the low-frequency 'blinking' PWM into the multi-kHz range where both people and cameras see steady light.

WHAT IS PWM DITHERING

Instead of doing all the dimming work in one long, slow cycle, the driver splits time into many tiny slices and spreads a little bit of extra on-time across them. The eye and a camera's exposure blends those slices together, so the result is the appearance of high frequency PWM without the visible flicker.

A QUICK EXAMPLE WITH IS31FL3248

Using Lumissil's IS31FL3248 as an example, this 48-channel LED driver supports either conventional PWM (8- or 16-bit) or selectable built-in temporal dithering (enhanced spectrum) with PWM splits such as 8+8 (for 16-bit look) or 8+4 (for 12-bit look) modes. For dithered modes the target resolution is expressed as B = H + L, where B is the desired PWM x-bit resolution and H + L represents the split dithered PWM bits. In dithered mode, the IS31FL3248 divides each frame into 2^L short segments; within each segment it runs an H-bit PWM, and it time-distributes the remaining L-bits across those segments so the time-average equals the full B-bit resolution. This maintains the requested B-bit accuracy while raising the per-segment PWM carrier to $\frac{f_{CLK}}{2^H}$ that is 2^L times higher than a single B-bit PWM frame, end result is a higher PWM frequency without giving up resolution. Note: The actual achievable PWM frequencies are still bound by the device's OSC capable setting/range, discussion to follow.

A concrete example for a conventional 16-bit resolution (which requires a slow PWM frequency) is to select the 8+8 dithered mode which splits the resolution to obtain a higher PWM frequency. The 8+8 dithered mode provides a "16-bit look" because the driver runs an 8-bit (H) high frequency PWM inside each short segment while distributing the remaining 8 bits (L) across $2^L = 256$ sub-segments. This is the "enhanced spectrum" mode: instead of the LED driver switching slowly to accommodate $2^{16} = 65{,}536$ time slots each PWM, it can quickly switch through $2^8 = 256$ small ones, recovering the full 16-bit look at a perceived higher multi-kHz PWM frequencies immune to LED flicker. In human terms: you're taking many small, fast sips of light rather than one long slow swallow, and the camera's exposure or human eye integrates these many small sips resulting in minimal flicker artifacts.

WHY DITHERED PWM LOOKS FASTER

With a single 16-bit PWM frame, the PWM frequency would be $\frac{f_{CLK}}{2^{16}} = \frac{f_{CLK}}{65{,}536}$ which due to the 2^{16} results in a very slow PWM frequency, regardless of the f_{CLK} oscillator frequency (e.g., at OSC= 16MHz the PWM=244Hz). In B=H+L dither mode, the frame is split into 2^L short segments and inside each segment, the PWM counter has 2^L counts, so the per-segment PWM frequency is $f_{PWM,seg} = \frac{f_{CLK}}{2^H}$. Depending on the PWM brightness value, the dominant perceivable PWM frame can range from $f_{frame} = \frac{f_{CLK}}{2^B}$, when only a few segments carry pulses and up to, when most or all segments are active. Time-averaging across segments recovers the full B-bit resolution "look" while shifting the dominant switching energy into the higher, camera-clean band set by $f_{PWM,seg}$.

To raise the PWM frequency while keeping 16-bit resolution, select the 8+8 dither mode. Here B = H + L; where B = "16-bit target resolution" and is divided into H=8-bit "PWM counter bits" and L=8-bit "number of segments".

For example, if a PWM $f_{CLK} = 16\,\text{MHz}$ is selected for a B=16-bit resolution using H+L=8+8 mode, then there will be 2^L=256 segments; with each segment PWM running at $\frac{f_{CLK}}{2^H} = \frac{16\,\text{MHz}}{2^8} = \frac{16\,\text{MHz}}{256} = 62.5\text{kHz}$, while the 16-bit macro-frame repeats at $f_{frame} = \frac{16\text{MHz}}{2^{16}} = 244.14\text{Hz}$. This greatly reduces visible LED flicker and rolling-shutter banding while keeping a true 16-bit average "look". Figure 1 shows a simplified "frame + gate" diagram showing which segments get a small extra on-time. The actual dither implementation is more like a MUX/Selector circuit so the normal PWM pulse in those selected segments is simply one count wider; there is no separate added pulse.

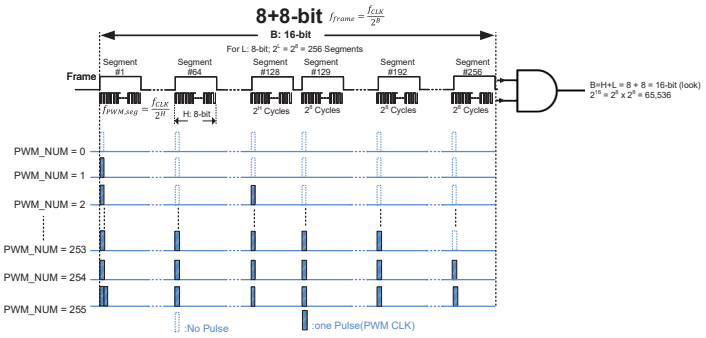

Figure 1. PWM Dithering for 16-bit Resolution in 8+8 Mode

Below is the comparison for a selected $f_{CLK} = 16MHz$ for conventional and dithered PWM:

- For a conventional 16-bit (B) resolution:

$$f_{PWM} = \frac{f_{CLK}}{2^B} = \frac{16\,MHz}{2^{16}} = \frac{16{,}000{,}000}{65{,}536} = 244.14\,Hz$$

- For 8+8 dither mode each segment PWM is:

$$f_{PWM,\,seg} = \frac{f_{CLK}}{2^H} = \frac{16MHz}{2^8} = \frac{16{,}000{,}000}{256} = 62{,}500Hz$$

- And to confirm, where B = H + L; 16-bit = 8-bit + 8-bit (dither):

$$f_{PWM,seg} = \frac{f_{CLK}}{2^H} = 2^L \frac{f_{CLK}}{2^B} = \frac{16{,}000{,}000}{2^{16}} = 256 \cdot 244.14Hz = 62{,}500\,Hz$$

The 2^L pushes the $\frac{f_{CLK}}{2^B}$ switching base frequency into the multi-kHz band, which is less likely to produce visible flicker or rolling-shutter bands on cameras.

PROGRAMMING STEPS

The examples below show how to program the IS31FL3248 for various dither modes. Follow these steps:

1. Table 9: Program the FC0 Configuration Register to select the desired dither mode PWMM [31:30]
2. Table 11: Choose an oscillator setting using FC1 Frequency Selection Register OSC [9:8]
3. Table 12: Select a PWM in the multi-kHz bracket based on the desired PWM dither mode
4. Table 3: Write the individual 48-channel brightness levels into PWM_H and PWM_L

Table 9 FC0 Unit (32:27) Configuration Register

Bit	32	31:30	29	28	27
Name	SSD	PWMM	AUTO_UPD	DTMRST	-
Default	0	00	0	0	1

Table 11 FC1 Unit (15:8) Frequency Selection Register and De-ghost & Spread Spectrum

Bit	15:14	13:12	11	10	9:8
Name	CLT	RNG	SSP	DE_GHOST	OSC
Default	00	00	0	0	00

Table 12 PWM Frequency

PWM Frequency	16MHz	8MHz	4MHz	2MHz
16-bit	244Hz	122Hz	-	-
8+8-bit	244Hz~ 62.5kHz	122Hz~ 32.2kHz	-	-
8+4-bit	3.9kHz~ 62.5kHz	1.95kHz~ 32.2kHz	0.98kHz~ 15.6kHz	0.49kHz ~7.8kHz
8-bit	62.5kHz	32.2kHz	15.6kHz	7.8kHz

Table 3 PWM0~PWM47 PWM Register

Unit	PWM47		…	PWM0	
Bit	767:760	759:752	…	15:8	7:0
Name	PWM_H	PWM_L	…	PWM_H	PWM_L
Default	0000 0000	0000 0000	…	0000 0000	0000 0000

IS31FL3248 8+4: 62.5KHZ/SEGMENT (FRAME 3.9KHZ; PERCEIVED FREQUENCY CONTENT ~3.9–62.5KHZ)

Objective: A 12-bit resolution "look" with multi-kHz PWM frequency by configuring the IS31FL3248 in H+L = 8+4 dither mode with OSC @ 16MHz.

Register Setup:

- Mode: PWMM = 10b (8+4 dither) in FC0.Configuration.
- Clock: OSC = 00b (16 MHz) in FC1 to place the per-segment PWM in the multi-kHz range (datasheet table 12 shows 3.9 kHz ~ 62.5 kHz at 16 MHz).
- Optional: Phase interleave: enable PHASE (FC1) so channel edges aren't simultaneous (lower ripple/EMI, less MLCC "singing").
- Optional: enable spread-spectrum (FC1: SSP/RNG/CLT) if you need to lower EMI peaks.

Example brightness (12-bit target = 0x34C):
Use PWM_H for the base and PWM_L[7:4] for the fractional nibble (driver spreads these over the 16 segments).

```
# Mode & timing
FC0.Configuration.PWMM  = 0b10  # 8+4 dither (16 time-
                                segments)
FC1.OSC                 = 0b00  # 16 MHz oscillator
#Optional: FC1.PHASE    = 1     # phase delay ON
#Optional:  FC1.SSP=0b1,  FC1.RNG=0b00,  FC1.CLT=0b00
#spread-spectrum

# Per-channel (0~47ch) PWM example (12-bit "look")
PWM_H[ch]          = 0x34    # base duty (8 MSBs)
PWM_L[ch][7:4]     = 0xC     # fractional (4 LSBs,
                             dithered over 24 = 16
                             segments)
PWM_L[ch][3:0]     = 0x0     # unused in 8+4
```

The final PWM duty cycle from PWM_H = 0x34 and PWM_L = 0x0C is: $\text{PWM duty} = \frac{0x34C}{2^{12}} = \frac{844}{4096} = 0.206\,(20.6\%\text{duty})$

CONCLUSION: FLICKER-FREE LED LIGHT

Lumissil's PWM dithering ("enhanced spectrum") delivers what conventional high bit resolution PWM can't: true 12–16-bit resolution at multi-kHz frequencies. By expressing resolution as B=H+L, the driver runs a fast H-bit PWM inside short time segments and time-distributes the L-bit "fraction" across 2^L segments. The time-average maintains the full B-bit resolution while the visible switching moves up by 2^L (e.g., 8+8 → 256×) producing LED light that appears flicker-free to people and reduces rolling-shutter artifacts on cameras light.