Application Note

Time Multiplexing : TPS92664-Q1, TPS92665-Q1, TPS92667-Q1 LED Matrix Managers

Jaron Wilson

ABSTRACT

This application note provides a comprehensive guide to implementing time multiplexing with Texas Instruments' third generation LED Matrix Manager (LMM) family: the TPS92664-Q1, TPS92665-Q1 and TPS92667-Q1. Time multiplexing allows one or more LMM devices controlling a series string of LEDs to share a single current source by sequentially activating subsets of LEDs over the PWM period. This technique applies both to systems with multiple LMMs in series and to single-device systems where each channel drives multiple LEDs in series. The technique enables size, cost and efficiency optimizations in multi-pixel automotive lighting applications. Practical hardware requirements, register configurations, EMI considerations, and example calculations are detailed.

Table of Contents

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

The TPS9266x LED matrix manager architecture partitions pixel-level LED control from the constant-current regulation stage using 16 internal shunt FETs to bypass individual LEDs. In traditional implementations each LMM-controlled LED string is supplied by the current source. However, headlamp real estate and budget restrictions motivate designs where a single current source drives two or more LMM-controlled LED strings connected in series. Time multiplexing achieves this by evenly dividing the on-time of each pixel within the PWM period. This minimizes the instantaneous forward voltage of the LED string ($V_{f,total}$) and verifies the maximum output voltage of the current source is not violated.

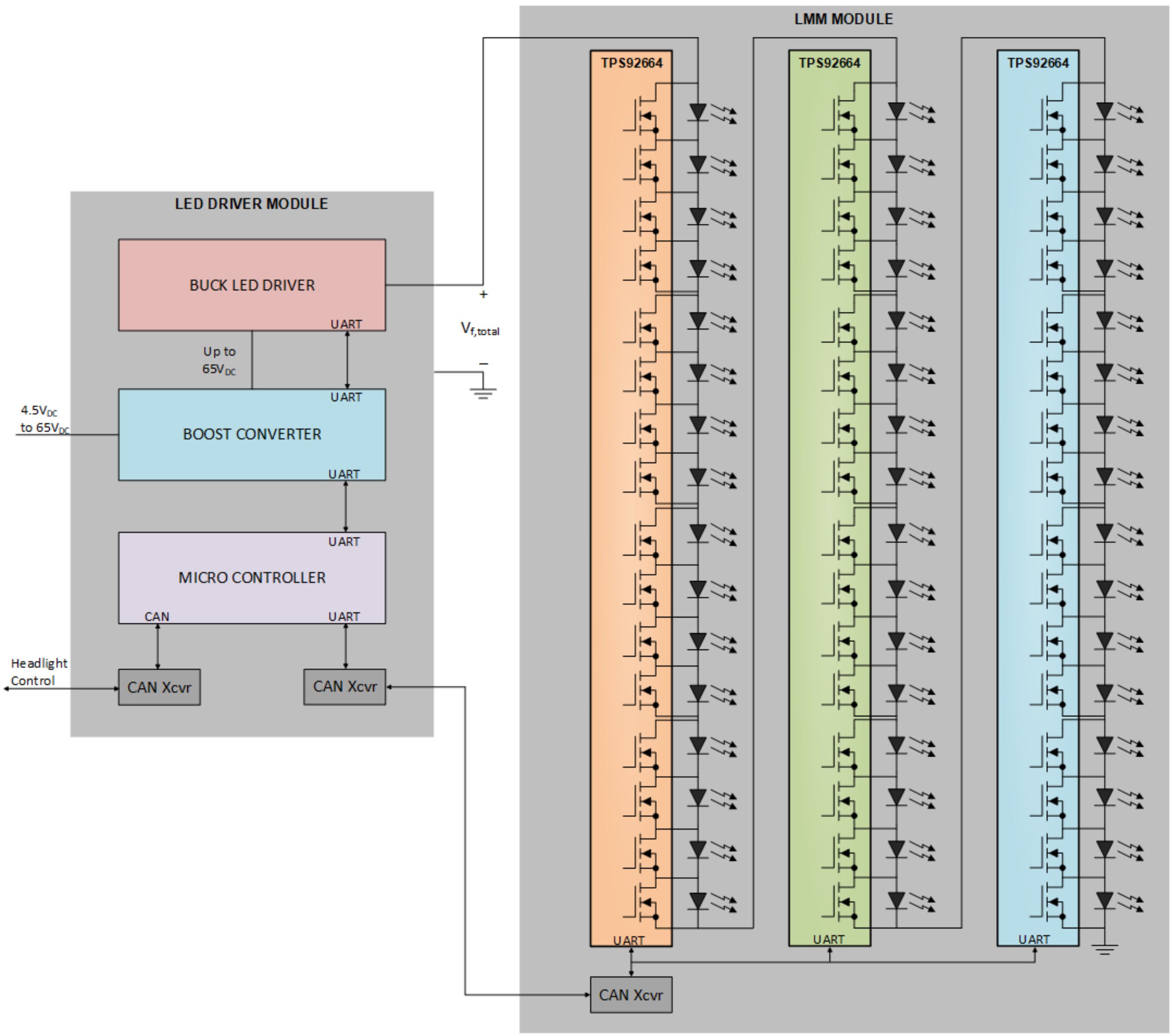

Figure 1-1. Three Device Time Multiplexing Block Diagram

While time multiplexing is often introduced in the context of extending the total LED string length across multiple LMMs, the same principle applies when a single LMM controls channels with multiple series-connected LEDs. In both cases, the purpose of time multiplexing is to reduce the instantaneous forward voltage of the total LED string so that this does not exceed the maximum output voltage of the current source.

The technique therefore serves as a general design for any configuration where $V_{f,total} > V_{max}$, regardless of whether the excess voltage arises from multiple LMMs connected in series or multiple LEDs per channel on a single device.

2 Time-Multiplexing Fundamentals

All sixteen internal bypass switches can be individually pulse width modulated (PWM) at a programmed frequency, phase-shift, and duty cycle. This shunt-FET dimming topology provides inherent phase shifting capability. The PWM clock is derived from a set of cascaded dividers (PTBASE and PTCNT located in the PWMTICK register) applied to the system clock. The system clock of the TPS92664 can be set to either 16MHz or 8MHz (1Mbps or 500kbps UART baud rate, respectively) by the CLKDIV2 bit located in the MTPCFG register (=0 for 16Mhz, =1 for 8Mhz). While the CLKDIV2 bit can be read in the MTPCFG register, the CLKDIV2 bit can only be written in the MTP_MTPCFG register. A new device has a default setting CLKDIV2 = 0, therefore first communication must always be at 1Mbps baud rate. If 500kbps communication is required, the CLKDIV2 bit must be set in the MTP_MTPCFG register and programmed into nonvolatile memory on all devices on the same UART bus. The TPS92665 can only be set to output 16MHz but can receive either 16MHz or 8MHz. The TPS92667 is a clock receiving device and does not have an internal oscillator.

ADDR	REGISTER	D7	D6	D5	D4	D3	D2	D1	D0	DEFAULT
0x82	PWMTICK	PTBASE[1:0]		PTCNT[5:0]						0001 1100

Bit	Field	Type	Reset	Description
7:6	PTBASE[1:0]	R/W	0x0	DIV1 Primary (base) CLK divider for PWM Clock
5:0	PTCNT[5:0]	R/W	0x1C	DIV2 Secondary CLK divider for PWM Clock

Figure 2-1. PWMTICK Register and Field Descriptions

Table 2-1. PTBASE Mapping of the Primary Divider (DIV1)

PTBASE[1:0]	DIV1
0 (default)	÷ 1
1	÷ 50
2	÷ 125
3	÷ 200

Table 2-2. PTCNT Mapping of the Secondary Divider (DIV2)

PTCNT[5:0]	DIV2	PTCNT[5:0]	DIV2	PTCNT[5:0]	DIV2	PTCNT[5:0]	DIV2
0	÷2	16	÷20	32	÷36	48	÷53
1	÷3	17	÷21	33	÷37	49	÷54
2	÷4	18	÷22	34	÷38	50	÷55
3	÷6	19	÷23	35	÷39	51	÷56
4	÷8	20	÷24	36	÷40	52	÷57
5	÷9	21	÷25	37	÷41	53	÷58
6	÷10	22	÷26	38	÷42	54	÷59
7	÷11	23	÷27	39	÷43	55	÷60
8	÷12	24	÷28	40	÷44	56	÷62
9	÷13	25	÷29	41	÷45	57	÷63
10	÷14	26	÷30	42	÷46	58	÷65
11	÷15	27	÷31	43	÷47	59	÷68
12	÷16	28	÷32	44	÷49	60	÷71
13	÷17	29	÷33	45	÷50	61	÷74
14	÷18	30	÷34	46	÷51	62	÷78
15	÷19	31	÷35	47	÷52	63	÷85

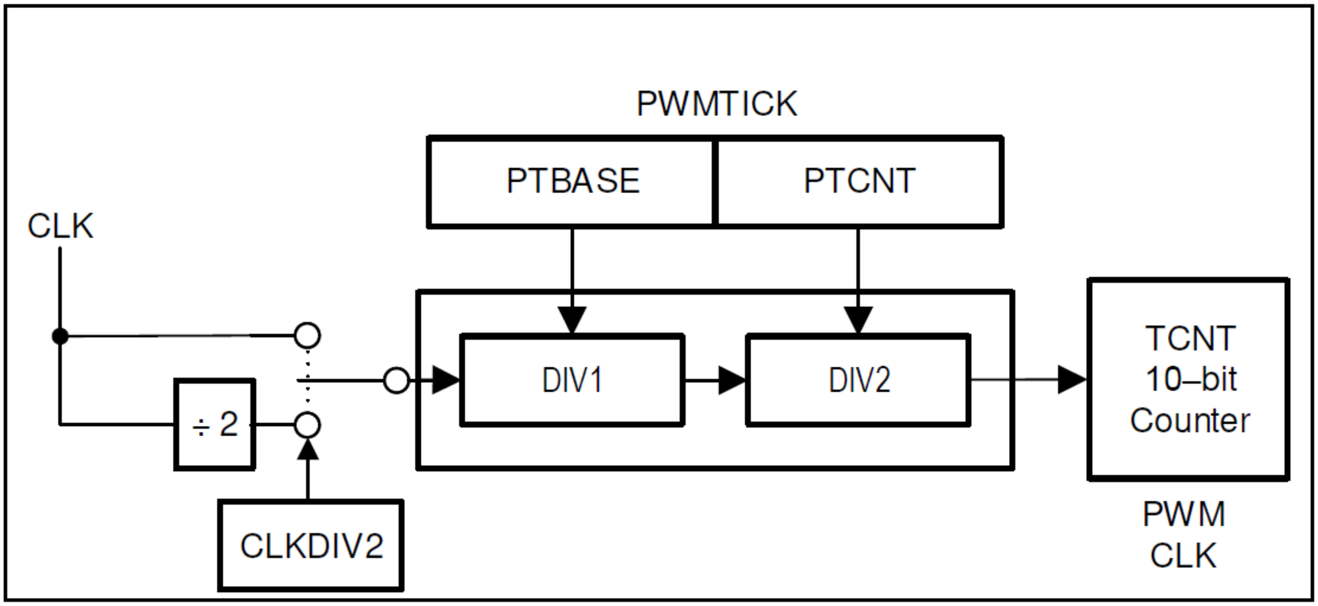

Figure 2-2. TPS92664 PWM Clock

$$PWM_{CLK} = CLK \div DIV1 \div DIV2 \quad (1)$$

PWM CLK Calculation Example: PWMTICK[7:0] = '00011100', CLKDIV2 = 0 (bit 7 from MTPCFG register, address (0x00h))

- PTBASE[1:0] = 0 → DIV1 = 1
- PTCNT[5:0] = 28 → DIV2 = 32
- CLKDIV2 = 0 → CLK = 16MHz
- PWMCLK = CLK ÷ DIV1 ÷ DIV2 = 16MHz ÷ 1 ÷ 32 = 500kHz

The PWM clock is used to generate a 10-bit counter (TCNT) that determines the PWM dimming frequency of the floating switches. Each PWM clock cycle, TCNT increments by one, starting from 0 and counting up to 1023, then starting again. 1024 counts of TCNT is one PWM period.

PWM frequency = 500kHz ÷ 1024 = 488Hz

WIDTHx registers control the pulse-width of the LED on-time (switch off-time), thereby controlling the duty cycle

$$\frac{WIDTH}{1024} = \text{Duty Cycle of the LED on time} \quad (2)$$

PHASEx registers control when in the PWM period the LED turns on or turns off depending on the value of the PSON bit in the SYSCFG register. Phase shifting occurs with respect to LED turn-off time when PSON = 0 or LED turn-on time when PSON = 1.

Therefore, when PSON = 0:

LED on-time begins when TCNT = PHASE – WIDTH

LED off-time begins when TCNT = PHASE

And, when PSON = 1:

LED on-time begins when TCNT = PHASE

LED off-time begins when TCNT = PHASE + WIDTH

Figure 2-3 shows a detailed example of PWM dimming (with PSON = 0). This is a general purpose example of PWM dimming, not specific to time multiplexing, simply to show how WIDTH and PHASE are implemented.

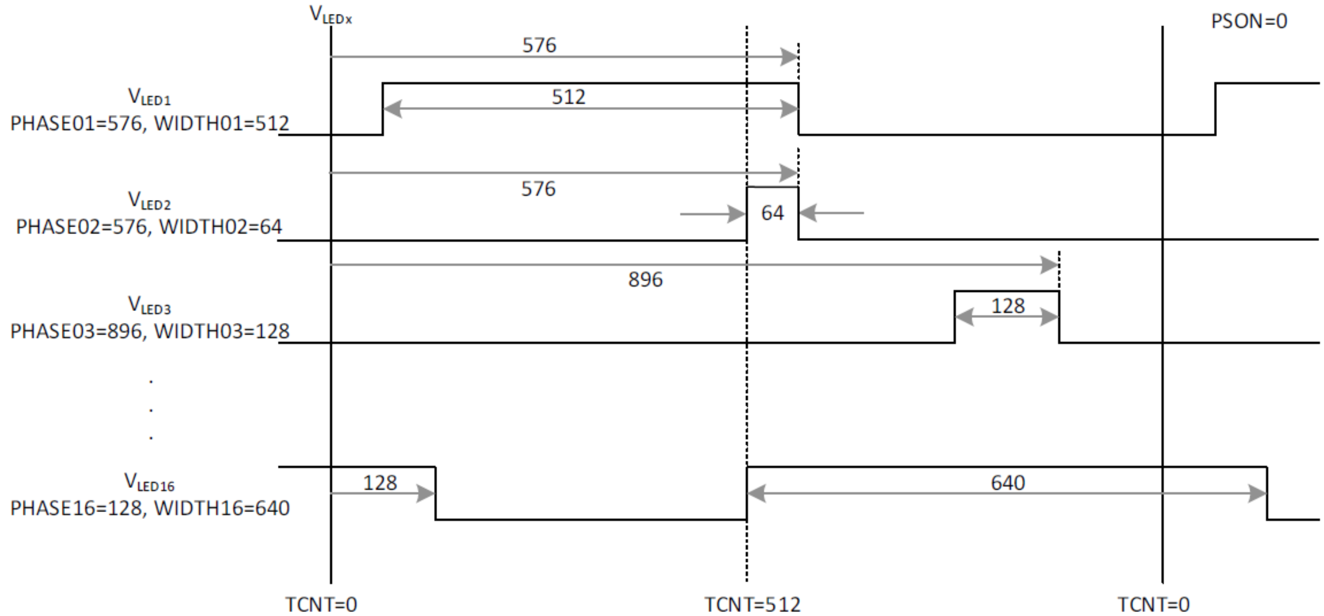

Figure 2-3. PWM Dimming with PSON = 0

Time multiplexing enables longer series strings of LEDs by carefully selecting PHASE and WIDTH values such that the entire LED string can be driven while the output voltage remains below the max output of the current source. This can be achieved by evenly spacing the on-time of each pixel throughout the PWM period. By evenly spacing the on-times of the pixel the minimum number of pixels are conducting at any given time, therefore minimizing the instantaneous forward voltage of the LED string. In order to evenly space the on-time of each pixel the PHASEx value increments must be set by:

$$\text{PHASExincrement} = 1024 \div \text{number of pixels multiplexed} \tag{3}$$

For example, if a user is multiplexing three LMMs, using all 16 switches on each device for a total of 48 pixels, then PHASEx = 1024 / 48 = approximately 21. Therefore:

PHASE01 = 0

PHASE02 = 21

PHASE03 = 42

PHASE04 = 63

To show the importance of PHASE shifting in managing the instantaneous forward voltage of the LED string there are two waveforms. They both plot the instantaneous forward voltage of three LMM LED string described above compared to the same maximum voltage output, however Figure 2-4 shows these waveforms without any phase shifting (all PHASEx = 0) and Figure 2-5 shows the effect of the equal phase shifting described above (increments of 21). The waveforms show that without phase shifting the forward voltage of the LED string far exceeds the maximum output voltage, however with even phase shifting the forward voltage of the LED string is comfortably within the maximum output voltage.

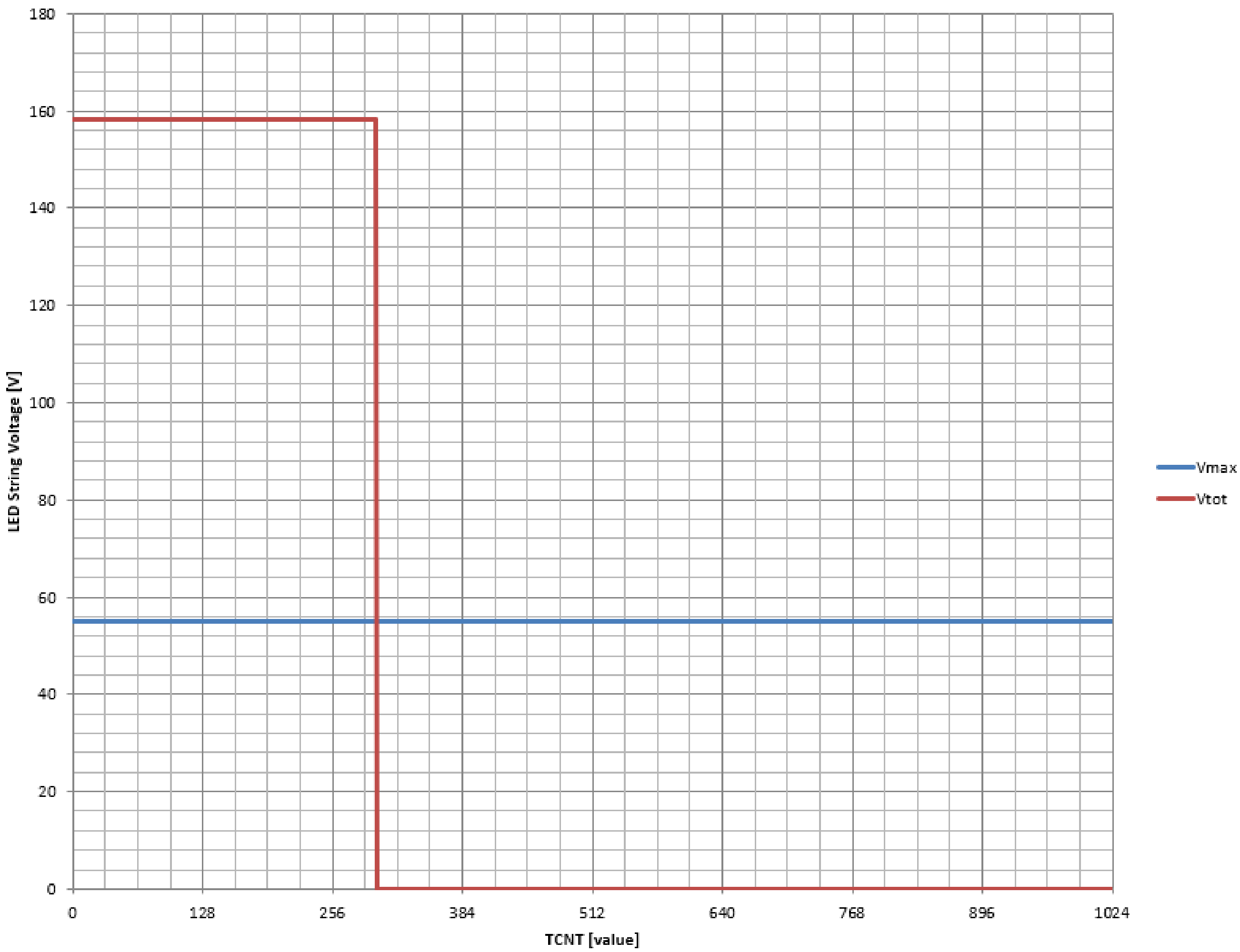

Figure 2-4. Voltage Waveforms With No Phase Shifting

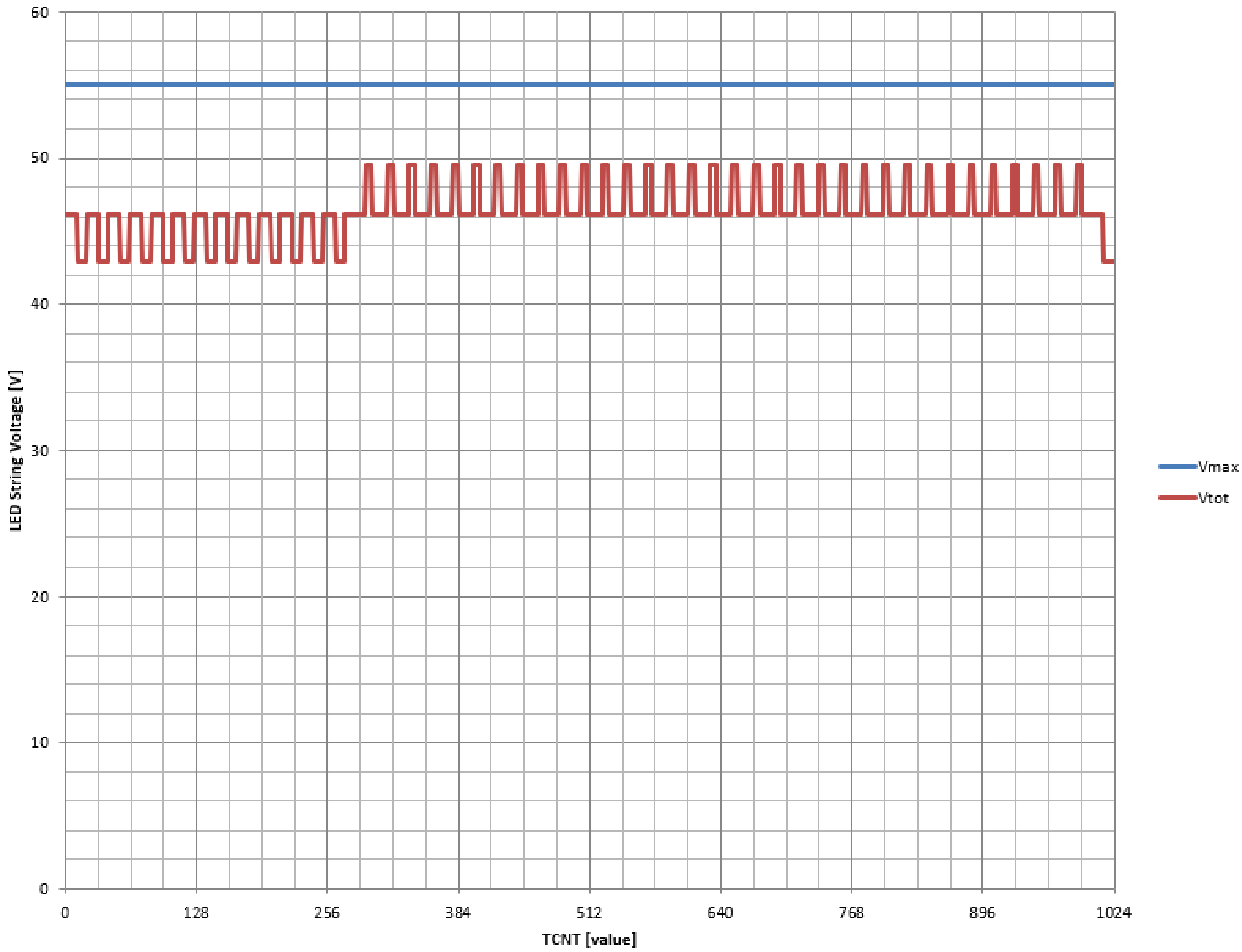

Figure 2-5. Voltage Waveforms With Optimal Phase Shifting

Time multiplexing cannot be implemented in every application as there are some critical requirements that must be met. The key requirements are:

- All LMMs share a common system clock and SYNC signals so the PWM generators stay synchronized.
- All devices must be programmed with the same PWMTICK value.
- When multiple LMMs are used, the LMMs must be located on the same PCB.
- WIDTHx values constrain the duty cycle so that the maximum output voltage of the current source is not violated.
- Nominal current through LEDs must be increased proportionally to maintain the same brightness at the lower maximum duty cycle (LED must be able to handle increased current)
- Instantaneous forward voltage of the LED string ($V_{f,total}$) during any TCNT must not exceed the current source maximum output voltage (V_{max}).

Time multiplexing can be applied in two primary scenarios:

1. Multi-Device Case: Extended Strings

   Multiple LMMs, each controlling one LED per channel, are connected in series to a single current source.
2. Single-Device Case: Multiple LEDs per Channel

One LMM drives multiple LEDs in series on each channel, producing a high total forward voltage without multiple devices.

In both architectures, PHASE and WIDTH settings distribute the conduction windows evenly throughout the PWM period to balance system voltage, optical output, and thermal load.

3 Hardware Design Guidelines

- Co-locate multiplexed LMMs on the same PCB as routing CLK_H/CLK_L signals through a harness do not pass EMI.
- Provide a continuous ground plane beneath CLK, SYNC, RX and TX traces to minimize capacitive coupling from high frequency traces.
- Terminate unused LVDS pairs with 100Ω across CLK_H/CLK_L on the last clock receiving device and two 47pF capacitors to GND from CLK_H/CLK_L on the clock generating device, see Figure 3-1.

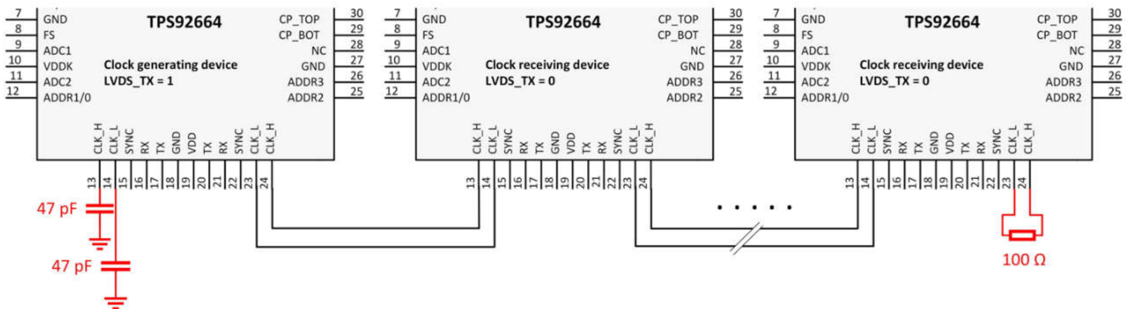

Figure 3-1. LVDS Termination Simplified Illustration

4 Register Configuration Procedure

1. Select system clock: the CLKDIV2 bit must be set in the MTP_MTPCFG register and programmed into nonvolatile memory on all devices on the same UART bus. Set CLKDIV2 bit =1 for 8MHz (500kbps) or =0 for 16 MHz (1 Mbps).
2. Program PTBASE and PTCNT in the PWMTICK register to set the desired PWM frequency.
3. Enable LVDS_TX on the clock generating LMM: set LVDS_TX bit in OUTCTRL register =1. Set LVDS_TX =0 on all clock receiving LMMs
4. Enable SYNC drive on the clock generating device: SYNCOEN and SYNCPEN bits in the OUTCTRL register = 1. All clock receiving devices set these bits to 0.
5. For every LED channel, write PHASEx such that:

$$\text{PHASEx} \quad \text{increment} = 1024 \div \text{number} \quad \text{of} \quad \text{pixels} \quad \text{multiplexed} \tag{4}$$

6. Use the Time Multiplexing Worksheet to determine WIDTHx values for every LED channel such that the LED string voltage does not exceed maximum output voltage (increase LED current as needed or tolerated to maintain nominal light output)
7. Program configuration in EEPROM if required (TPS92664 only, see MTP programming flow).

5 MTP Programming Flow

Below are the steps necessary to program the EEPROM on the TPS92664 with a provided example:

1. Write to the desired **MTP** register at the MTP DEVID address. Note this address is different from the volatile DEVID address, see Figure 5-1 for mapping. In this command, the DEVID byte is coming from the *MTP* column in Figure 5-1.

DEVID[3:0] Address set by ADDRx Pins	DEVID[7:0] Byte	
	MTP	VOLATILE
Decimal	Hex	Hex
0	0x80	0x20
1	0xC1	0x61
2	0x42	0xE2
3	0x03	0xA3
4	0xC4	0x64
5	0x85	0x25
6	0x06	0xA6
7	0x47	0xE7
8	0x08	0xA8
9	0x49	0xE9
10	0xCA	0x6A
11	0x8B	0x2B
12	0x4C	0xEC
13	0x0D	0xAD
14	0x8E	0x2E
15	0xCF	0x6F

Figure 5-1. Volatile and Non-Volatile Address Map

2. Burn the EEPROM by writing the *program* code to the **volatile** programming registers (beginning at MTP_PROG1) with a 4 byte write with the data: CA 23 35 24. In this command your DEVID byte is coming from the *volatile* column in the previous table. Note if a user is programming multiple devices this step must be done as a *broadcast write*, for example, using DEVID = 0xBF.
3. Then if a user power cycles the device, the corresponding volatile register is loaded with the value in the MTP register you just programmed as the new default.

Below is an example of this flow with commands provided, where device address 0's PWMTICK register is programmed with the value 0x1C and burned into the device EEPROM as the default.

1. So, to perform a single byte write (INIT = **0x87**) to MTP device address 0 (DEVID = **0x80**) to the MTP_PWMTICK register (REGADDR = **0x07**) with (DATA = **0x1C**) then the command frame with CRC is: **87 80 07 1C 2A A5**
2. Then, the user must burn the EEPROM by writing the code to the volatile programming registers (beginning at MTP_PROG1) with a 4 byte write with the data: CA 23 35 24

   So, to perform a 4 byte write (INIT = **0xAA**) to the volatile device address (DEVID = **0x20**) to the MTP_PROG1 register (REGADDR = **0xFB**) with the hex programming code (DATA = **0xCA 0x23 0x35 0x24**), the hex command frame with CRC is:

   AA 20 FB CA 23 35 24 31 E7
3. Power cycle the device and verify that the desired value is now loaded in the PWMTICK register.

6 Spreadsheet-Based Voltage Analysis

The provided Excel tool helps to visualize the total forward voltage of the LED string compared to the maximum voltage output. There are two sections of the Time Multiplexing Worksheet: input and output. In the input section all of the cells that should be edited by the customer are in the upper left-hand corner of the sheet and highlighted in yellow. The inputs are: number of LMM's being multiplexed, maximum voltage output of the current source, maximum voltage of the LED pixel, WIDTH, and PHASE. The PHASE input has an auto-calculate option which, if checked, automatically sets the PHASE to be evenly spaced according to the number of LMMs being multiplexed (assuming all 16 switches are used on each device). The auto-calculate formula is =1024 / (# of LMM × 16), rounded to the nearest whole number. If the auto-calculate formula is left unchecked, then the customer can simply enter the desired PHASE value into the same cell.

# of LMM	**2**	*number of LMMs being multiplexed										
MAX V	**55**	*maximum voltage output of buck converter (must be <=60V)										
VLED	**3.3**	*maximum voltage of LED "pixel" (pixel is defined by a single matrix switch in parallel)										
				1st TPS92664 Device								
WIDTH	**500**			500	500	500	500	500	500	500	500	500
PHASE	**32**	auto-calculate?	☑	0	32	64	96	128	160	192	224	256
LEDOFF				500	532	564	596	628	660	692	724	756

Figure 6-1. Time Multiplexing Worksheet Inputs

The user must enter inputs in accordance with the application and the spreadsheet calculates the total forward voltage of the LED string for every TCNT 0-1023. The same spreadsheet methodology applies to both multiplexing across multiple LMMs and multiplexing channels within a single LMM. In both cases, PHASE increments and WIDTH selection follow identical rules, the only difference is the maximum voltage of the LED pixel increases when there are multiple series LEDs across one switch.

After the user has finished entering their inputs, the user can view the two waveforms plotted over the course of the PWM period. First is the V_{max} waveform. This is plotted in blue and is equal to the maximum output voltage that the customer entered on the *Input* tab. This waveform is constant throughout the PWM period and determined by the customers' choice of current source.

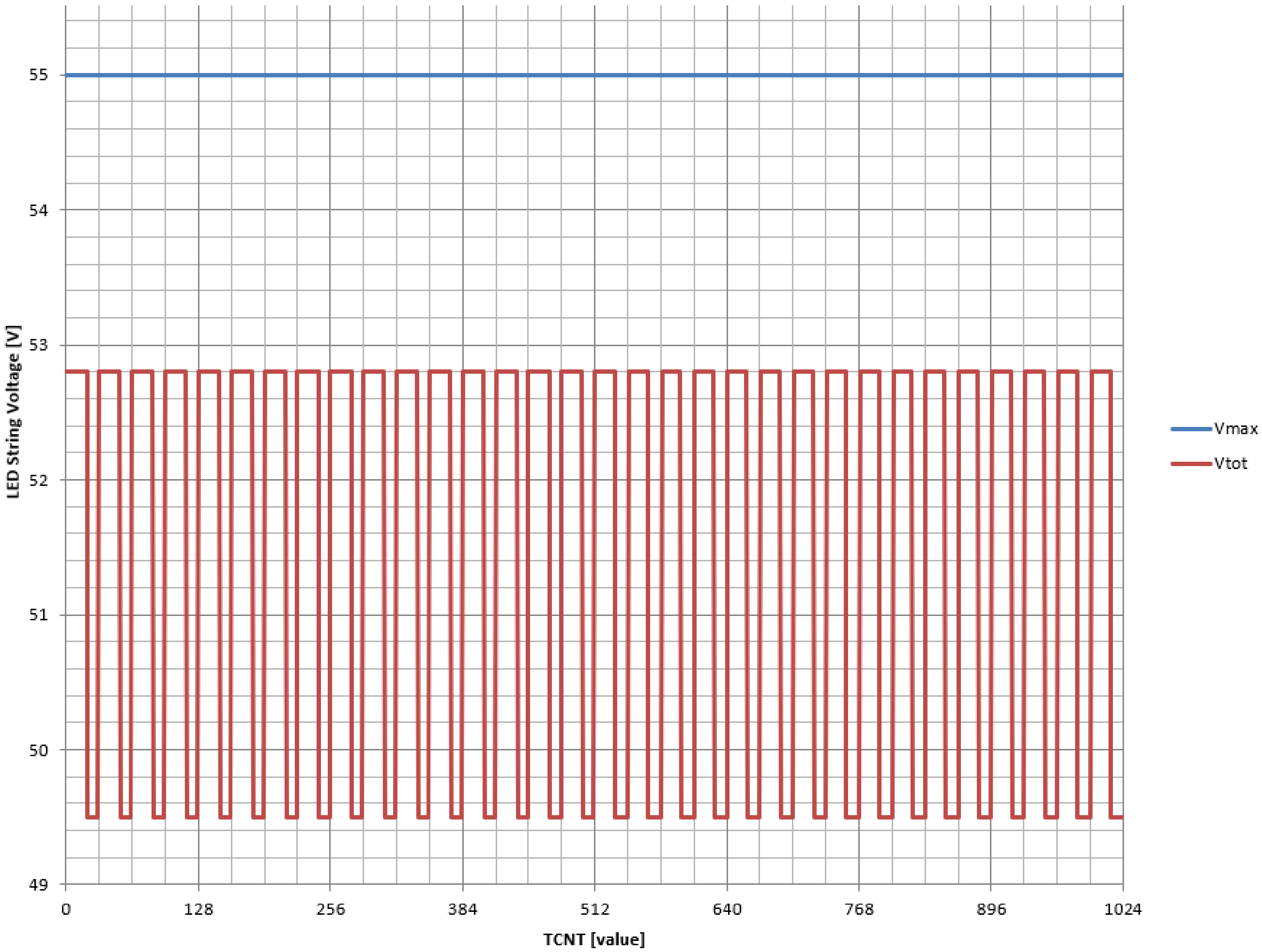

Figure 6-2. Time Multiplexing Worksheet Output

The waveform plotted in red, $V_{f,total}$, is the total forward voltage of the LED string and is equal to the $V_{f,total}$ column on the "input" tab. In order to not violate the maximum output voltage of the current source the $V_{f,total}$ waveform must remain below the V_{max} waveform for the entirety of the PWM period. There must also be some amount of headroom between $V_{f,total}$ and V_{max} to account for process variation, temperature fluctuation, and so on. Note that there is a resulting increase in LED current to maintain the same nominal level of brightness expected. For example, if the Time Multiplexing Worksheet reveals a 33% maximum duty cycle, then LED current ,must be increased 3x for the same light output. Of course, the LEDs selected must also be able to tolerate this increase in current. Customers are encouraged to modify inputs and check the waveforms to develop a better intuitive understanding of how the variables affect the total string voltage.

7 Example Design 1: Two TPS92664-Q1 Devices on One Current Source

Below you can find an example of a typical time multiplexed application consisting of 2x TPS92664 devices controlling a series string of 32 LEDs (1 LED/switch) driven by a 55V maximum current source.

- Number of LMMs Multiplexed = 2
- 32 LEDs, V_{f_max} = 3.2V at 105°C
- Current source: V_{max} = 55V

Given these variables, the PHASEx values must increment in steps of 32 since:

1024 / 32 = 32 (PHASEx increment= 1024÷number of pixels multiplexed)

So, the selected PHASE values are:

Table 7-1. Example PHASEx Values

Device	PHASEx	Value
0	PHASE01	0
0	PHASE02	32
0	PHASE03	64
0	PHASE04	96
0	PHASE05	128
0	PHASE06	160
0	PHASE07	192
0	PHASE08	224
0	PHASE09	256
0	PHASE10	288
0	PHASE11	320
0	PHASE12	352
0	PHASE13	384
0	PHASE14	416
0	PHASE15	448
0	PHASE16	480
2	PHASE01	512
2	PHASE02	544
2	PHASE03	576
2	PHASE04	608
2	PHASE05	640
2	PHASE06	672
2	PHASE07	704
2	PHASE08	736
2	PHASE09	768
2	PHASE10	800
2	PHASE11	832
2	PHASE12	864
2	PHASE13	896
2	PHASE14	928
2	PHASE15	960
2	PHASE16	992

Simulation with the spreadsheet tool confirms that with WIDTH values of 500 (approximately 48% duty cycle) yields $V_{f,total}$ peaks at 51.2V (< 55V), leaving some headroom for transients and variation.

# of LMM	2	*number of LMMs being multiplexed
MAX V	55	*maximum voltage output of buck converter (must be <=60V)
VLED	3.2	*maximum voltage of LED "pixel" (pixel is defined by a single matrix switch in parallel)

			1st TPS92664 Device																2nd T
WIDTH	500		500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500
PHASE	32	auto-calculate? ☑	0	32	64	96	128	160	192	224	256	288	320	352	384	416	448	480	512
LEDOFF			500	532	564	596	628	660	692	724	756	788	820	852	884	916	948	980	1012

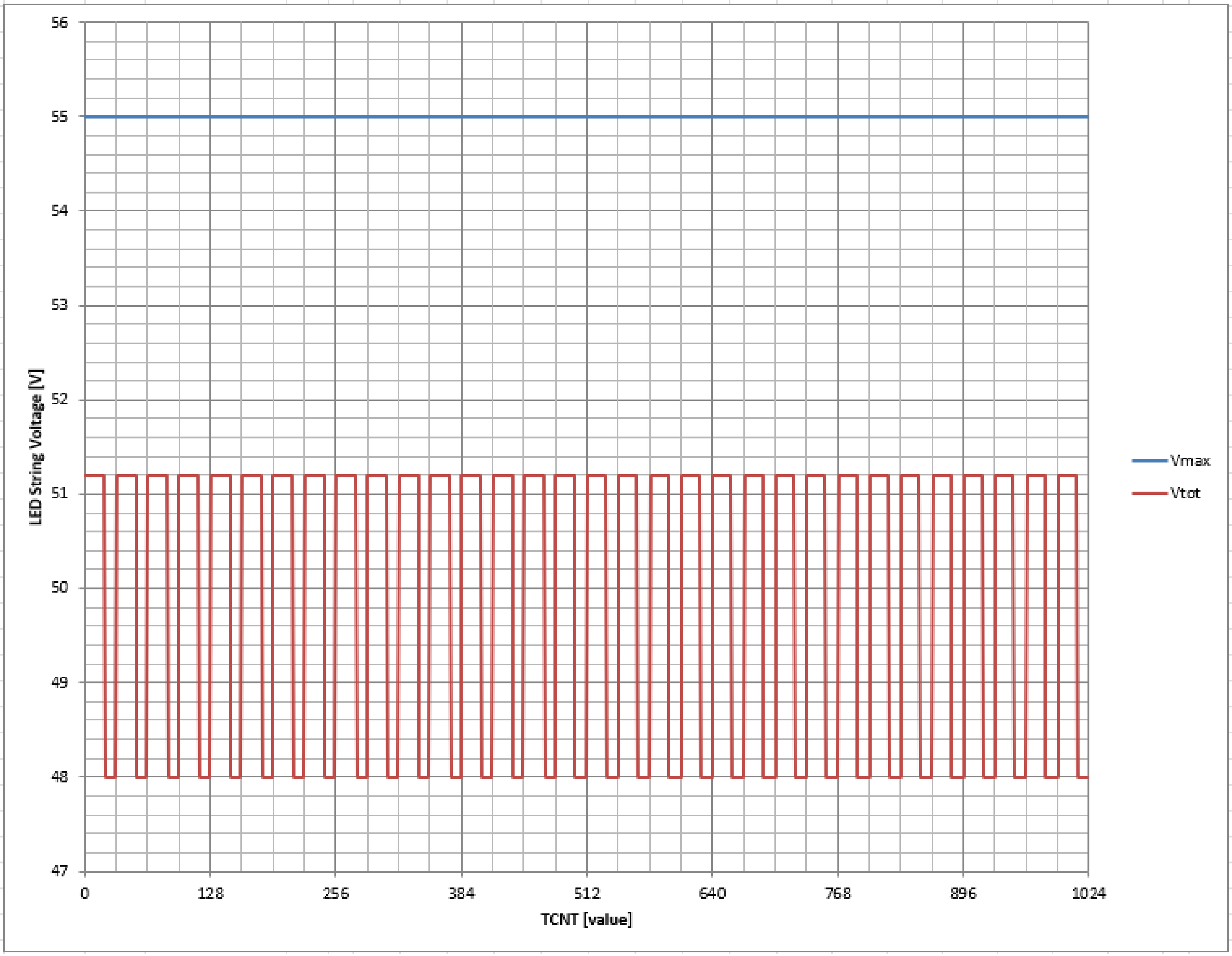

Figure 7-1. Example Design 1 Spreadsheet

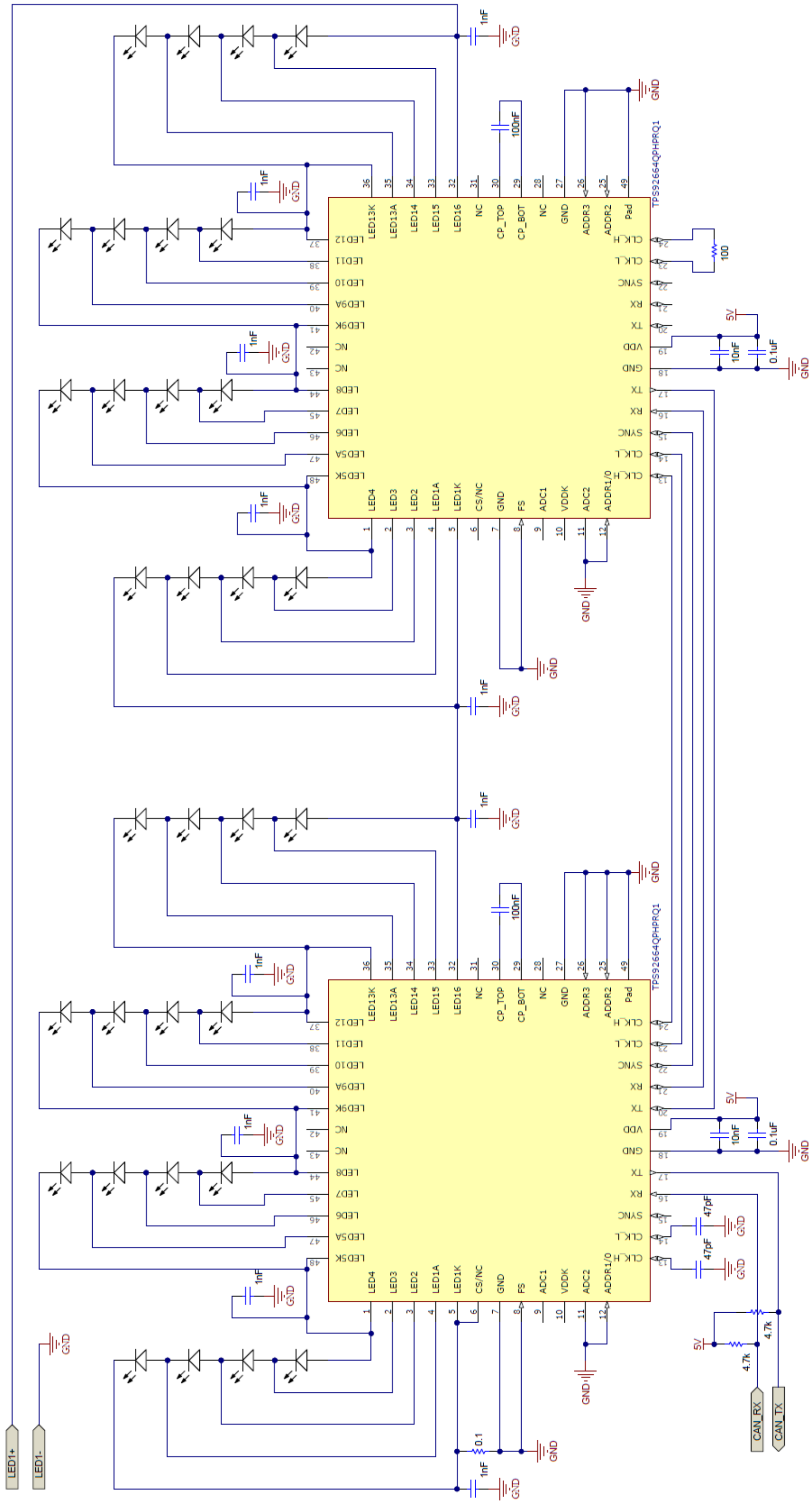

Figure 7-2. Example Application with Multiple Devices

8 Example Design 2: Single TPS92664-Q1 with Multiple LEDs per Channel

In this configuration, one TPS92664-Q1 controls sixteen channels, each driving two LEDs in series (Vf ≈ 6.6V per channel). The potential total forward voltage of all channels conducting simultaneously reach approximately 105.6V, exceeding the 55V compliance of the current source.

To maintain operation under 55V, the sixteen channels are evenly spaced throughout the PWM period. PHASE values are spaced by 64 counts (1024 ÷ 16), and WIDTH is selected so that the summed instantaneous $V_{f,total}$ per phase remains about 50V.

# of LMM	1	*number of LMMs being multiplexed
MAX V	55	*maximum voltage output of buck converter (must be <=60V)
VLED	6.6	*maximum voltage of LED "pixel" (pixel is defined by a single matrix switch in parallel)

			1st TPS92664 Device																2nd
WIDTH	500		500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	0
PHASE	64	auto-calculate? ☑	0	64	128	192	256	320	384	448	512	576	640	704	768	832	896	960	0
LEDOFF			500	564	628	692	756	820	884	948	1012	52	116	180	244	308	372	436	0

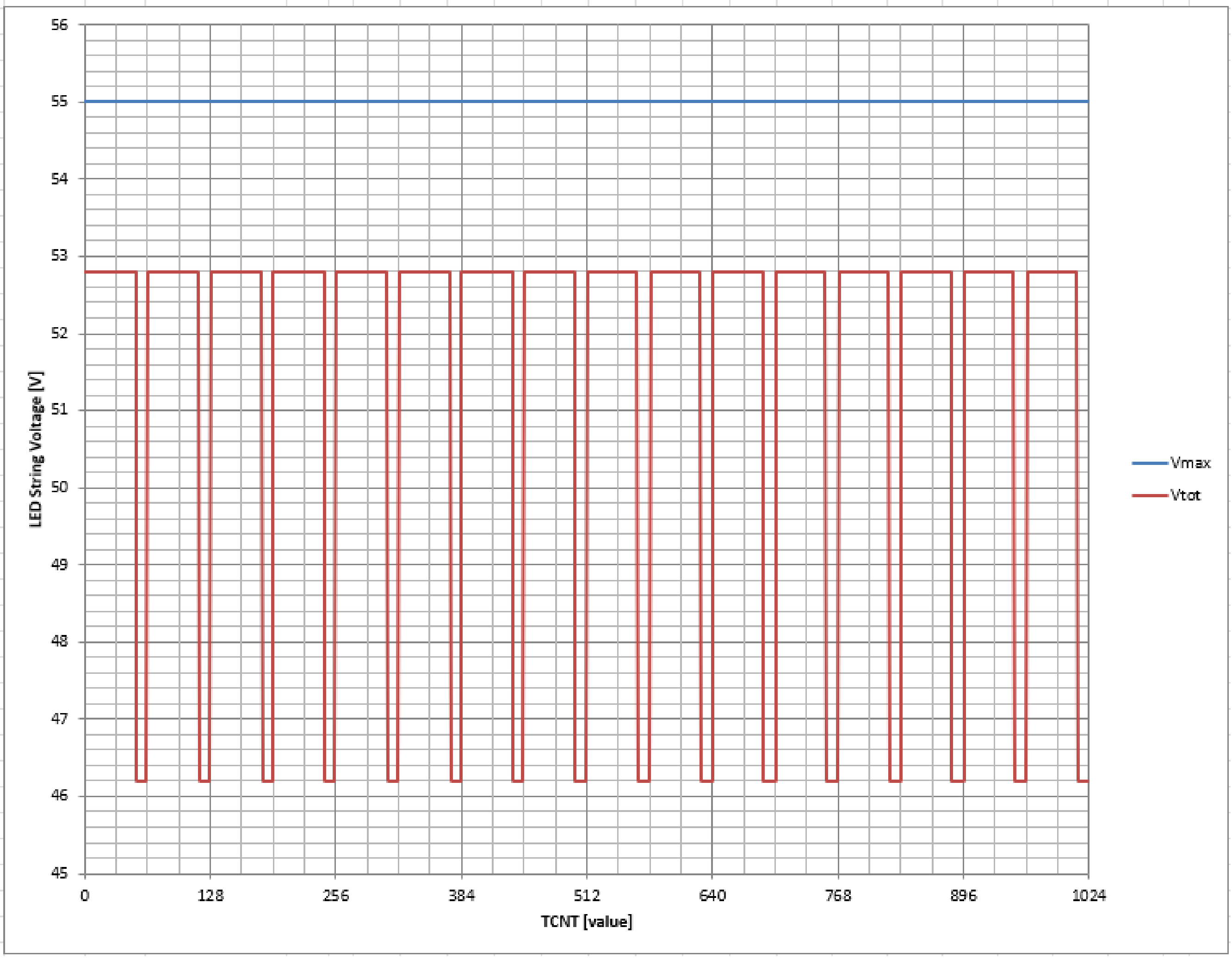

Figure 8-1. Example Design 2 Spreadsheet

This single-device example demonstrates that time multiplexing can limit instantaneous voltage even without multiple LMMs, simply by scheduling channel activity in staggered time slots.

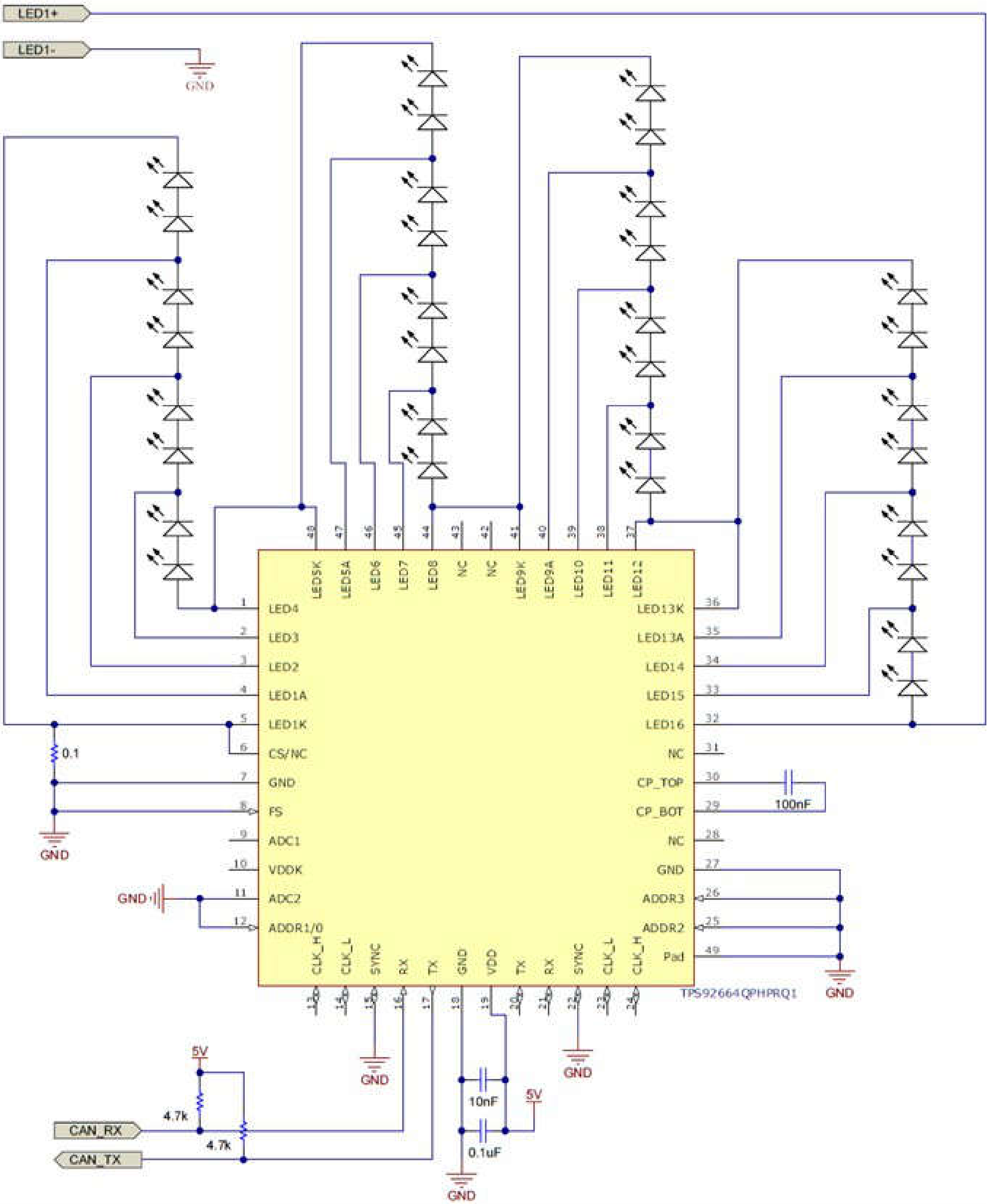

Figure 8-2. Example Application With Single Device

9 Summary

Time multiplexing with the TPS92664, TPS92665, and TPS92667 offers a practical way to reduce the number of current regulators in multipixel lighting systems without sacrificing control or light quality. By aligning device clocks, synchronizing PWM generators, and selecting PHASE and WIDTH values to distribute conduction evenly, designers can keep the LED string voltage within regulator limits while maintaining system efficiency. The provided spreadsheet tool simplifies this process by allowing quick verification of voltage margins across the PWM cycle.

Time multiplexing is not limited to multidevice LED strings. This can also be applied within a single LMM when each channel drives multiple series LEDs. In both architectures, the goal remains the same: maintain compliance with the current source voltage limit while optimizing cost and utilization efficiency.

When implemented with proper PCB layout, LVDS termination, and EEPROM programming, time multiplexing delivers a repeatable and robust design that scales simply to different lighting architectures. The example design demonstrates how these methods can be directly applied, giving customers confidence in building compact, cost-optimized, and reliable lighting design with the TPS9266x family.

10 References

- Texas Instruments, *TPS92664-Q1 Automotive Low Noise 16-Channel LED Matrix Manager with Advance Diagnostics, Integrated Oscillator, and EEPROM*, datasheet.
- Texas Instruments, *TPS92665-Q1 Automotive Low Noise 16-Channel LED Matrix Manager with Advance Diagnostics and Integrated Oscillator*, datasheet..
- Texas Instruments, *TPS92667-Q1 Automotive Low Noise 16-Channel LED Matrix Manager with Advanced Diagnostics*, datasheet.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL 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 AND 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 skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale, TI’s General Quality Guidelines, or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.