Please note that Cypress is an Infineon Technologies Company.

The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content

The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers

Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

THIS SPEC IS OBSOLETE

Spec No: 001-52581

Spec Title: IMPLEMENTING WHITE LED DRIVER USING PSOC(R) WITH AMBIENT LIGHT SENSOR INTERFACE - AN52581

Sunset Owner: Rajiv Vasanth Badiger (RJVB)

Replaced by: None

Implementing White LED Driver Using PSoC® with Ambient Light Sensor Interface

Author: Rajiv V Badiger
Associated Project: Yes
Associated Part Family: CY8C27443
Software Version: PSoC Designer™ 5.0
Related Application Notes: AN52491

If you have a question, or need help with this application note, contact the author at rjvb@cypress.com.

Abstract

Most LCD display systems require backlighting and an ambient light sensor interface. This application note presents two techniques using PSoC® to drive white LEDs, along with interfacing an ambient light sensor without external processing. This provides a low cost and small size solution that is crucial for any portable system.

Contents

Introduction

Today, portable systems contain various sensors along with LCD and keypad backlighting. Sensors require some analog processing while white LEDs used in LCD and keypad backlighting requires drivers. PSoC offers a single subsystem that can interface to sensors, perform signal conditioning, and also offer a closed loop control system for driving backlights.

This application note explains the strategies to drive six white LEDs connected in series and interfacing ambient light sensor to control the backlight intensity. It discusses the following:

- Inductor boost circuit basics
- Inductor boost implementation using PSoC with
  - Hardware method
  - Firmware method
- Comparison of implementation types
- Automatic backlight intensity control using ambient light sensor

The appendices at the end of the document give:

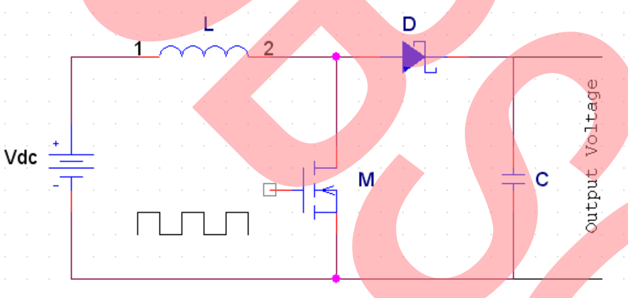
- Project information
- Resource consumption in various PSoC parts
- Test setup to measure efficiency

Boost Circuit Basics

In LCD backlighting, series or parallel combinations of white LEDs are used. Each LED has forward voltage drop of 2.8 V to 3.6 V. If connected in series, the voltage required to drive up to six LEDs is high compared to source voltage. However, in portable systems, the available voltage ranges from 1.2 V to 3.7 V. This cannot directly drive the series connected white LEDs. For this reason, boost networks are used.

The boost network is made using inductor and N Channel MOSFET switching arrangement as shown in Figure 1.

Figure 1. Simple Boost Network

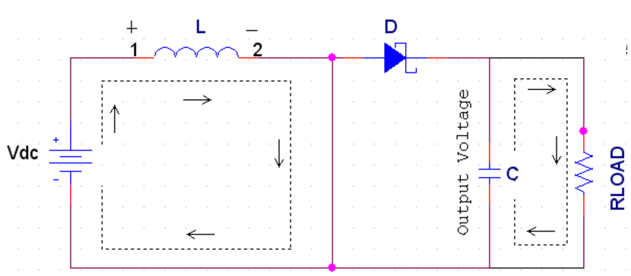


MOSFET M is continuously switched ON and OFF using a high frequency PWM signal.

Consider a case when the MOSFET is ON. This causes current to flow from source, inductor, and MOSFET as shown in Figure 2. The node 1 is at a positive potential when compared to node 2. During this half cycle, current through the inductor rises slowly from its initial value—typical inductor property!

Energy is stored in the inductor during this half cycle.

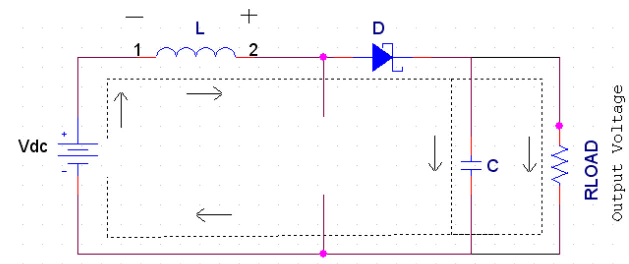
Figure 2. Case 1: MOSFET in ON State



Consider the next half cycle of the PWM, when it is logic zero. This causes MOSFET to be turned OFF. But the inductor does not allow current to change abruptly; again typical inductor property! The only path for the current to flow is through Schottky diode and output capacitor and load. The impedance of this output network is much more than the MOSFET ON state resistance. To force the same current, the inductor develops higher voltage in reverse direction (now node 2 is positive with reference to node 1). It slowly releases energy through the output branch. During this cycle output capacitor builds up voltage.

This storing and releasing of energy repeat to maintain nearly constant voltage at the output.

Figure 3. Case 2: MOSFET in OFF State

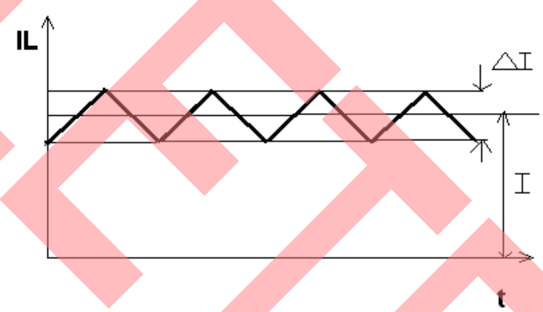


The boost network can be operated in two modes:

- Continuous Conduction Mode (CCM)
- Discontinuous Conduction Mode (DCM)

In CCM, frequency and boost circuit component values are selected such that current through the inductor never becomes zero for all value of load currents. In this mode, output voltage depends only on the duty cycle of the PWM driving the MOSFET. Figure 4 shows the current wave shape in CCM.

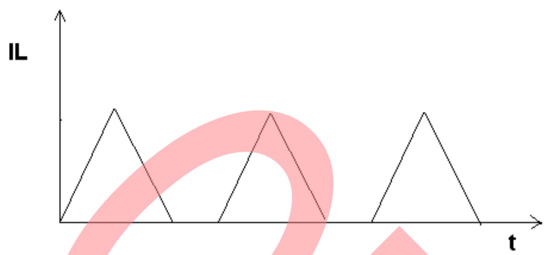
Figure 4. Inductor Current Waveform in CCM



Current I is the average load current; ripple ΔI represents charge and discharge cycle of the output capacitor current.

In DCM, current through the inductor becomes zero for a specific duration in the PWM cycle. This mode comes into effect for light loads. This is understood from the CCM waveform. If current I becomes smaller than ripple current ΔI, it makes inductor current zero for some time, as shown in Figure 5. In this mode, output voltage depends on the duty cycle and the output load impedance.

Figure 5. Inductor Current Waveform in DCM



DCM is used for larger boost ratios. However, in DCM mode, the inductor current has larger peak values. This increases core losses in the inductor. It can also create more EMI issues and are used for low power applications.

But DCM relaxes constraints on the inductor value and switching frequency selected. Lower switching frequency and inductor value results in DCM mode operation. This helps to reduce at least the switching losses, which are more in CCM mode due to higher operating frequency. In this implementation, discontinuous conduction mode is chosen.

To implement backlight LED driver, the same boost network is used. However, white LEDs must be driven with constant current, irrespective of variations in voltage supply or any other parameter such as temperature. Therefore, providing only the PWM signal to the boost network does not help in maintaining constant current. This requires a closed loop system.

Along with boosting and maintaining constant current, the network must also offer these features:

- Over voltage protection (open circuit)
- Over current protection (short circuit)
- MOSFET current limit
- Soft start

This application note explains how these features are implemented using PSoC.

Inductor Boost Implementation

With the programming and configurable hardware features, PSoC can be made to drive white LEDs using:

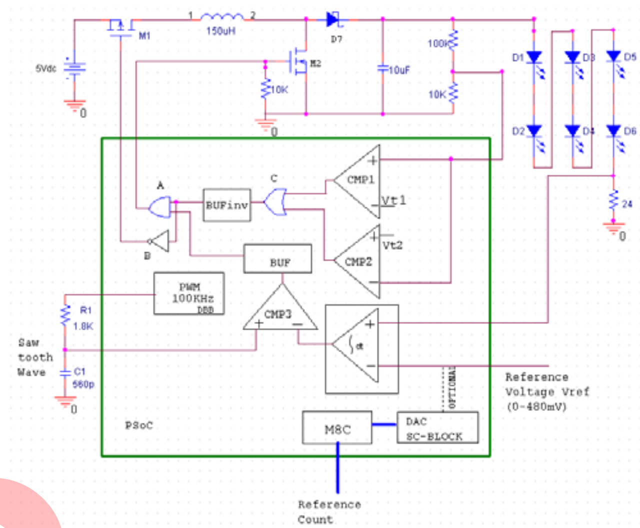
- Hardware method
- Firmware method

Hardware Based Backlight Control

This method does not involve any firmware action to control the LED current. It relieves M8C (PSoC processor) from controlling the boost network output.

The block diagram in Figure 6 illustrates the concept of creating constant current source that drives the LEDs.

Figure 6. Hardware Based Backlight Control



The complete system consists of boost network, output LED array, and control system made using PSoC. As shown in Figure 6, boost network needs inductor, MOSFET, and the Schottky diode to be connected externally.

The boost network uses a 150 µH inductor and lower frequency of 100 kHz. This results in a DCM operation. The system also uses one more MOSFET M1 (PMOS) to switch off the power to boost network when any short circuit or open circuit event happens.

The control system uses the following major blocks:

- Difference integrator
- PWM generator (100 kHz)
- Comparator (CMP3)
- DAC (optional)
- Comparator (CMP1): over voltage protection
- Comparator (CMP2): over current protection

All these are created inside PSoC. Only the square to saw tooth wave converter, which is created with RC components, is outside. The design is not critical for this integrator, because the slope of the saw tooth does not make any difference in its working. A signal which is a periodic unique time-continuous function is all that is required.

The current feedback is taken through 24 Ω resistance connected in series with LEDs. One resistive potential divider is connected across the output of boost network to

sample the output voltage to detect short and open circuit events.

Working

The LED current is set by adjusting the reference voltage at the difference integrator non inverting input. The feedback voltage, proportional to the LED current is fed to the inverting input of difference integrator. The integrator output is given by:

$$K1 * \int K2 * (Vfb - Vref)dt \qquad \text{Equation 1}$$

Integrator output rises if the difference is positive, $Vfb > Vref$ and falls if the difference is negative, $Vfb < Vref$.

This difference integrator output is used to set the trigger level for the comparator CMP3, which receives saw tooth wave as another input. Thus the output of CMP3 is a PWM signal whose duty depends on the trigger level set, that is the integrator output.

As integrator output builds up or falls, PWM duty cycle varies, thereby changing the boost output voltage.

As an example, consider that actual current is less than the reference set. The integrator output falls, decreasing the trigger level of CMP3. This increases the duty cycle of the PWM driving the boost circuit, thereby increasing the output voltage. It leads to increase in LED current. This current again causes change in the integrator output. The process continues as long as actual current is less than the reference set.

Finally, when the actual current feedback is equal to the reference value, integrator output stabilizes, keeping the PWM duty cycle constant. Figure 7 illustrates this operation.

Figure 7. Hardware Based Backlight Intensity Control

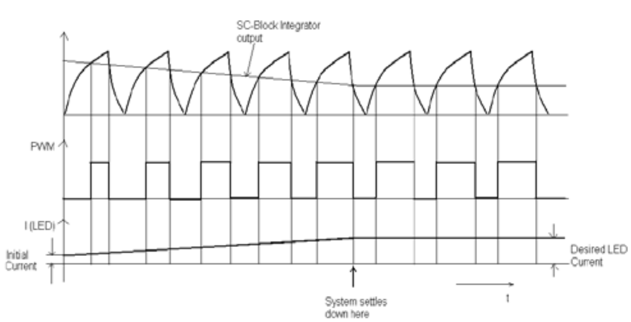


Figure 7 is only for the purpose of explanation. In reality, it takes many integration cycles before the system arrives at steady state.

LED current is given by the following equation:

$$I(LED) = \frac{Vfb}{24\Omega} = \frac{Vref}{24\Omega} \qquad \text{Equation 2}$$

Reference voltage can be set from 0 to 480 mV; this sets the current from 0 to 20 mA. The reference voltage is given either by an external analog voltage source or by using an internal DAC.

If a DAC is used to generate the reference current value, then the resolution in current control is limited. DAC can generate a maximum of REFHI (2.6 V in this case). A 6-bit DAC is used for the application which gives the resolution as $\frac{2.6V}{64} = 0.04V$.

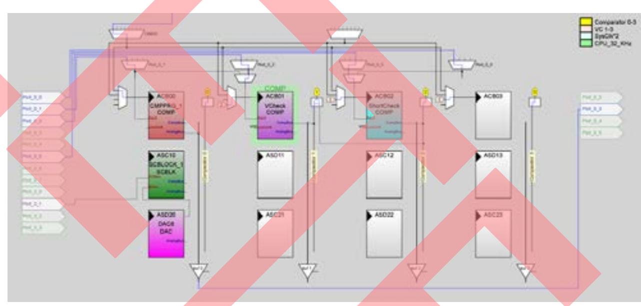
So current can be varied in steps of $\frac{40mV}{24} = 1.66mA$.

This gives the maximum number of dimming levels possible as $\frac{20mA}{1.66mA} = 12levels$

Module Placement and Routing

Analog Section

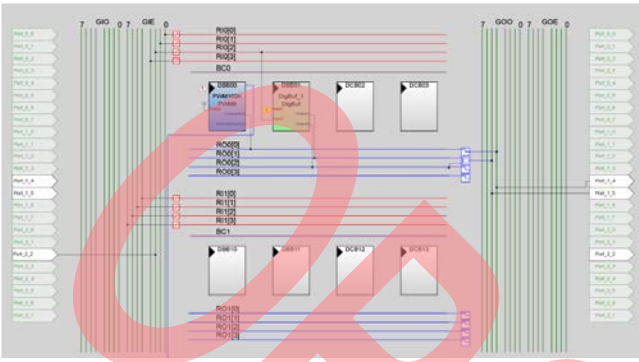
Figure 8. Module Placement - Analog



CMPPRG_1	CMP3
VCheck	CMP1
ShortCheck	CMP2
SC_BLOCK_1	SC Block Integrator
DAC6	DAC

Digital Section

Figure 9. Module Placement - Digital



PWM100K	PWM
DigBuf_1	BUFinv, BUF

The PWM is used to generate saw tooth wave is also used as a sampling clock for the difference integrator made using SC block.

Protection Features-

This system employs protection networks for the following:

- **Over voltage (open circuit):** As shown in Figure 2, resistive potential divider is connected across the output to take the sample of boost output voltage.

  It is connected to comparator CMP1, which receives Vt1 as the threshold voltage. When the output LED branch is open, the voltage at the output capacitor rises to an unsafe level. If no protection is offered, this can damage the Schottky diode and MOSFET. When output capacitor voltage exceeds the threshold, the output of CMP1 goes high, which is taken to DigBuf (inverting mode). The output of this buffer is then fed to an AND gate A which drives MOSFET gate. Thus AND gate A output is disabled when overvoltage event happens, thereby disabling the boost circuit that supplies current to the LEDs.

  Also the PMOS switch M1, connected to the supply in series is disabled when this event happens.

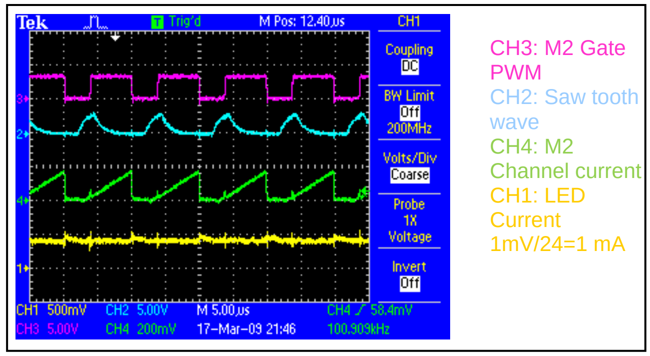
- **Over current (output short circuit):** This event refers to complete short at the boost output. To detect this event again, voltage at the output is observed. If it falls below the threshold, CMP2 output goes high, which turns OFF the output of AND gate A.

  This also turns the input PMOS switch M1 off, shutting down the supply to boost network.

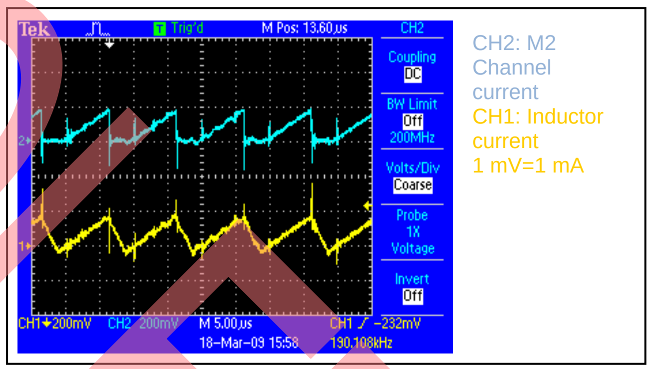
- **Soft start feature**: This feature is essential to avoid sudden inrush in inductor current when the system is powered ON. There is a surge in inductor current when duty cycle of the PWM is high on startup. When the board is powered, firmware takes control to increase the reference gradually, thus eliminating initial inrush of current.

Signal Captures

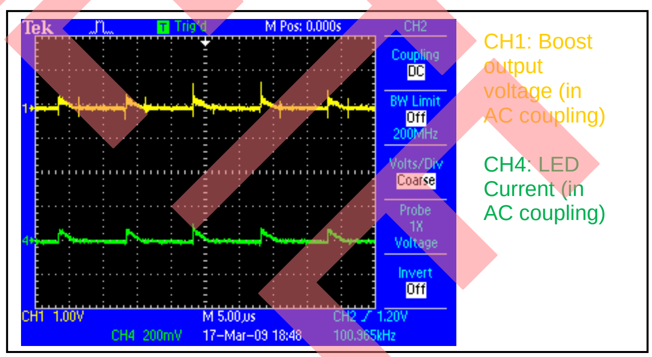
Operating waveforms at 18 mA LED current



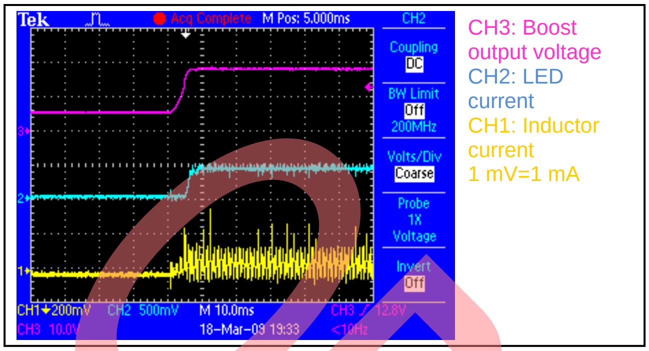
Input current at 18 mA LED current



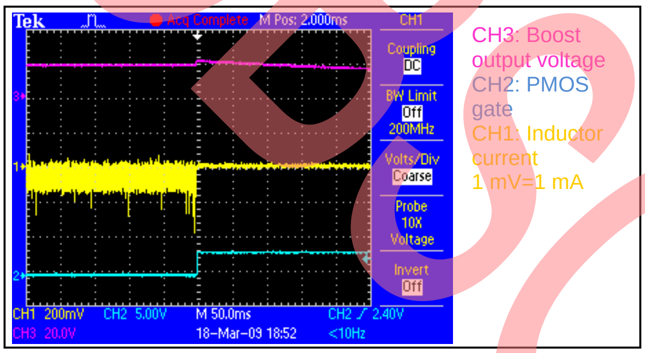
Ripple Content at 18 mA LED current



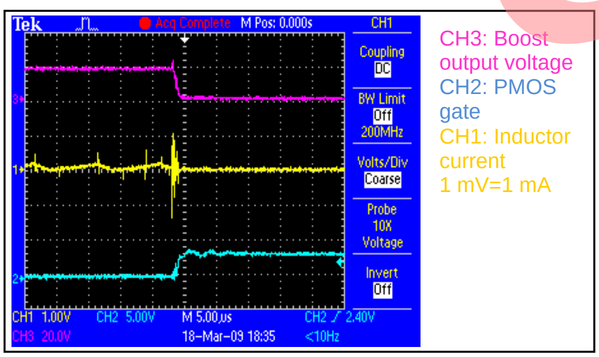
Soft start at18 mA final LED current



Open circuit

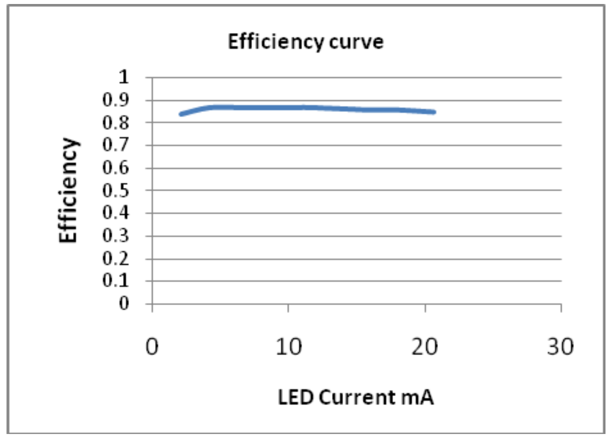


Short circuit



Efficiency

Figure 10. Efficiency of Boost Network



This shows efficiency of only boost network and does not include power consumed by the device.

See Appendix C for the test setup used to measure efficiency.

Table 1. Selecting an Inductor

Inductor Value	Minimum Operating Voltage	Peak Inductor Current
33 µH	3 V	430 mA
100 µH	4.2 V	260 mA
150 µH	4.8 V	180 mA
220 µH	5 V	160 mA

Test condition: Current feedback is kept within ±10 mV of the set current value.

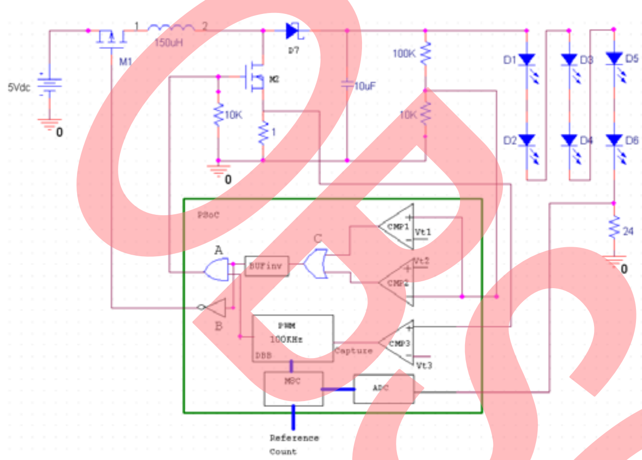
Table 2. Specifications

Parameter	Value
PWM Frequency	100 kHz
Maximum Duty Cycle	75%
Operating Voltage Range	4.8 V to 12 V (for 150 µH)
Maximum LED Current	20 mA
Dimming Type	Analog
Overvoltage Threshold	20.6 V
Short Circuit Threshold	2.31 V
Soft Start (Settling Time)	40 ms
Resolution in Current Setting	1.69 mA (when 6 bit DAC is used)

Firmware Based Backlight Control

This method involves controlling the PWM duty through firmware action based on feedback from ADC, which measures the current value.

Figure 11. Firmware Based Backlight Control



It uses the same boost circuit, but there is no need to square to saw tooth converter here.

Internally, it uses:

- ADC
- CMP1 – for open circuit protection
- CMP2 – for output short protection
- CMP3 – for MOSFET current limit
- PWM

Working

ADC measures the LED current value, forming the feedback loop. The error detector role of the control system is performed by firmware.

The system receives reference value from the user and compares it with the present current value. If current is less than the reference set, PWM duty cycle is increased to adjust the current to that reference. On the other hand, duty cycle of the PWM is decreased, if the system finds more LED current than the reference set.

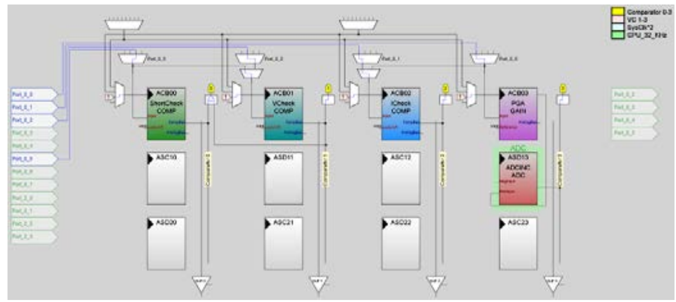
An 8-bit ADC is used in the project. It maps 0 to 2.6 V input voltage from 0 to 255 counts.

As the current sample resistance is set to 24 Ω, the feedback voltage at 20 mA is 480 mV. This gives the ADC count as 0 to 47 for 0 to 20 mA LED current. Reference count is set between 0 and 47.

Module Placement and Routing

Analog Section

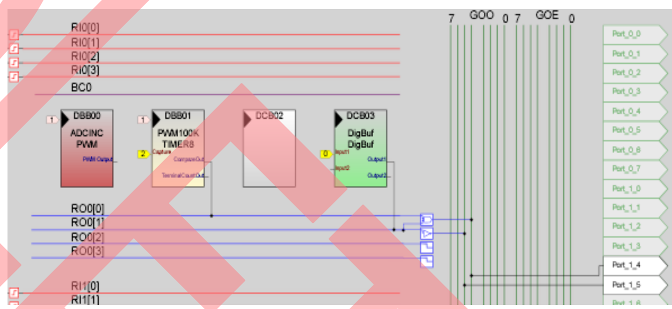
Figure 12. Module Placement - Analog



ADCINC	ADC
CMP1	VCheck
CMP2	ShortCheck
CMP3	ICheck

Digital Section

Figure 13. Module Placement - Digital



Note that PWM100K is implemented using TIMER8 module. The Timer module has a feature called Capture, which is useful for implementing MOSFET current limit feature.

Protection Features

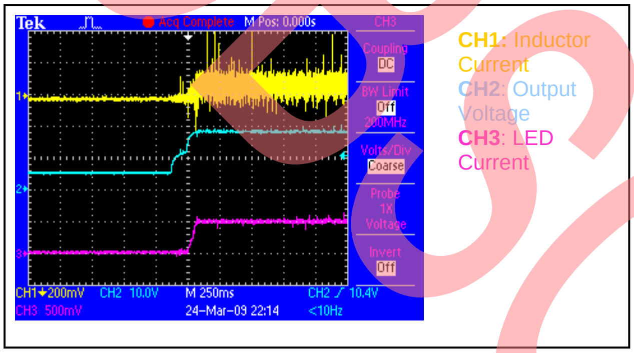
- **Open circuit protection**: Refer open circuit protection of hardware based backlight control.
- **Output short circuit protection:** Refer output short circuit protection of hardware based backlight system.
- **MOSFET over current protection:** In Figure 7, 1 Ω resistance is connected to source terminal of MOSFET. This provides the MOSFET channel current feedback to the control system. CMP3 provides this protection. When the current through MOSFET exceeds 625 mA (when MOSFET is ON and PWM is high), CMP3 triggers the capture of PWM100K. This inputs the present count into the PWM Duty register, bringing PWM line at logic 0. This limits the time for

which MOSFET is ON. When a capture event happens, the PWM attains logic 0 state, providing instant protection. Also, reference count is decremented by '1' in firmware, to check the MOSFET current for lower output LED current. Thus the system tries to adapt to the given situation.

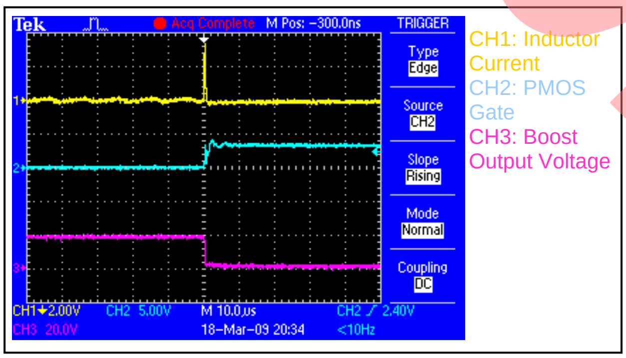
- **Soft start:** This feature is essential to avoid sudden inrush in inductor current when system is powered ON. There is a surge in inductor current when the PWM duty cycle is kept high on startup. In firmware, the reference count is incremented gradually to limit the initial inrush of current.

Signal Captures

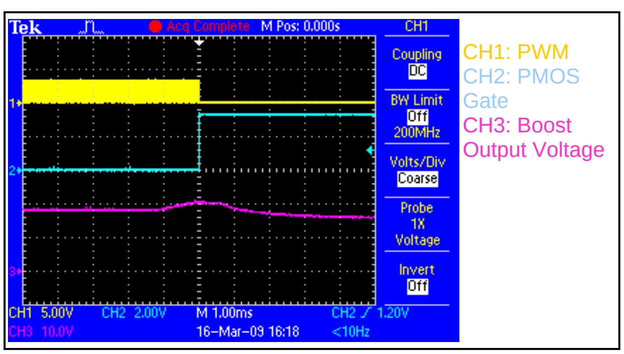
Soft start



Short circuit protection



Open circuit protection



Selecting inductor value based on minimum voltage and peak current:

Table 3. Values for 20 mA LED Current

Inductor	Minimum Voltage	Inductor Peak Current
33 µH	2.5 V	450 mA
100 µH	2.8 V	280 mA
150 µH	3 V	200 mA
220 µH	3.3V	180 mA

Test condition: Current feedback is kept within ±10 mV of the set current value.

Efficiency

Efficiency is similar to that of the hardware based method, shown in Figure 10, because the same external components are used.

Table 4. Specifications

Parameter	Value
PWM Frequency	100 kHz
Duty Cycle Limit	91%
Soft Start Time	250 ms
Operating Input Range	Min 3 V for 150 µH
Maximum LED Current	20 mA
Dimming	Analog
Over Voltage Threshold	20.6 V
Short Circuit Threshold	2.31 V
MOSFET Current Limit	625 mA
Resolution in Current Setting	0.4 mA ±0.4 mA (probable error)

Which Method to Select?

You can implement either of the methods based on requirements and resources available.

The hardware based control is suitable when better accuracy in current control and very less CPU overhead is required.

For a miniature solution, go for firmware based control.

Backlight System with Ambient Light Sensor

Ambient Light Sensor (ALS) plays a key role in system performance by reducing unnecessary power consumption in the display power.

ALS is used to sense the surrounding light conditions and modify the backlight LED brightness (both LCD and keypad backlight). For bright surrounding conditions, LCD display brightness requires a high level to enable the user to see display properly. On the other hand, non display elements such as keypad do not require backlight under high ambient light conditions, so keypad backlight can be shut down.

In a dark environment, LCD brightness must be reduced, otherwise it becomes bright for the naked eye to handle. In this condition, backlighting for keypad must be switched ON.

The goal is to make the LCD brightness same as ambient light.

Table 5. Use of ALS

Backlighting	Dark	Bright
LCD	Less	More
Keypad	ON	OFF

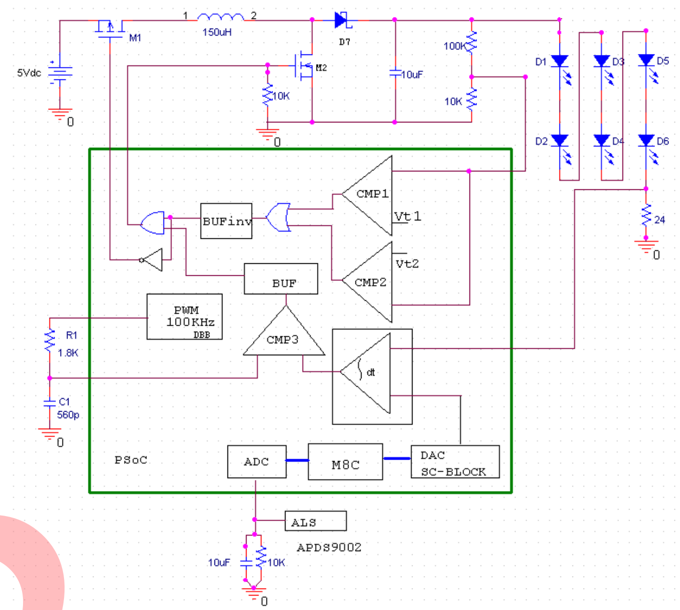
Integrating ALS with Backlight System

The ALS sensor chosen for this purpose is APDS9002.

For more details on interfacing ALS sensor with PSoC, see Cypress application note AN52491.

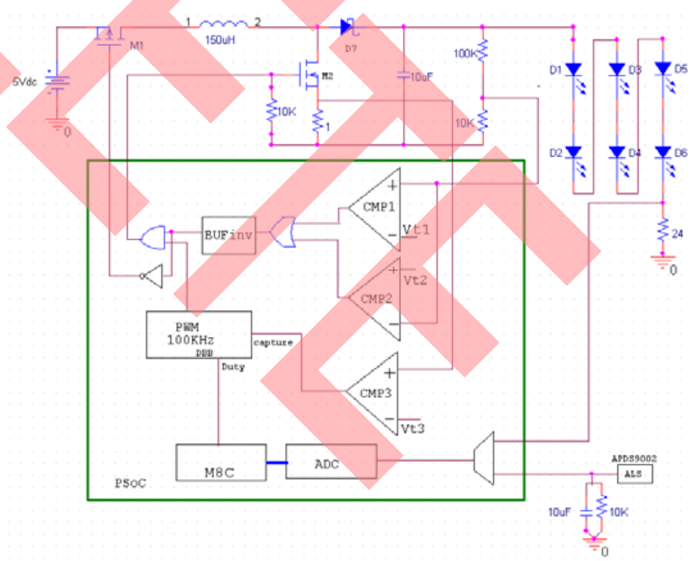
In the hardware control method, ALS value is read through ADC. An additional module ADC must be introduced into the solution. ADC measures ALS value and controls the reference voltage through DAC.

Figure 14. Solution for Backlight Control with ALS (Hardware Method)



The firmware method already uses ADC to measure LED current. The same ADC can be multiplexed to measure the ALS value and control the LED current reference value. There is no additional resource requirement to implement this solution.

Figure 15. Solution for Backlight Control Using ALS (Firmware Method)



You can either provide steps in backlight intensity or continuous variation. In real applications, it is not desirable to change backlight intensity instantly when ALS value changes. PSoC's programming features allow introducing delay in changing the backlight intensity and ensuring the ambient light has changed and is at a steady value. This can bring more integration and reduce cost of the complete solution

Summary

With the programming and configurable hardware features, PSoC provides flexibility in implementing the backlight solution. Also, this is not the only subsystem that you can make using a single chip; there can be many more inside.

About the Author

Name:	Rajiv Badiger
Title:	Applications Engineer
Contact:	rjvb@cypress.com

Appendix A

Project Information

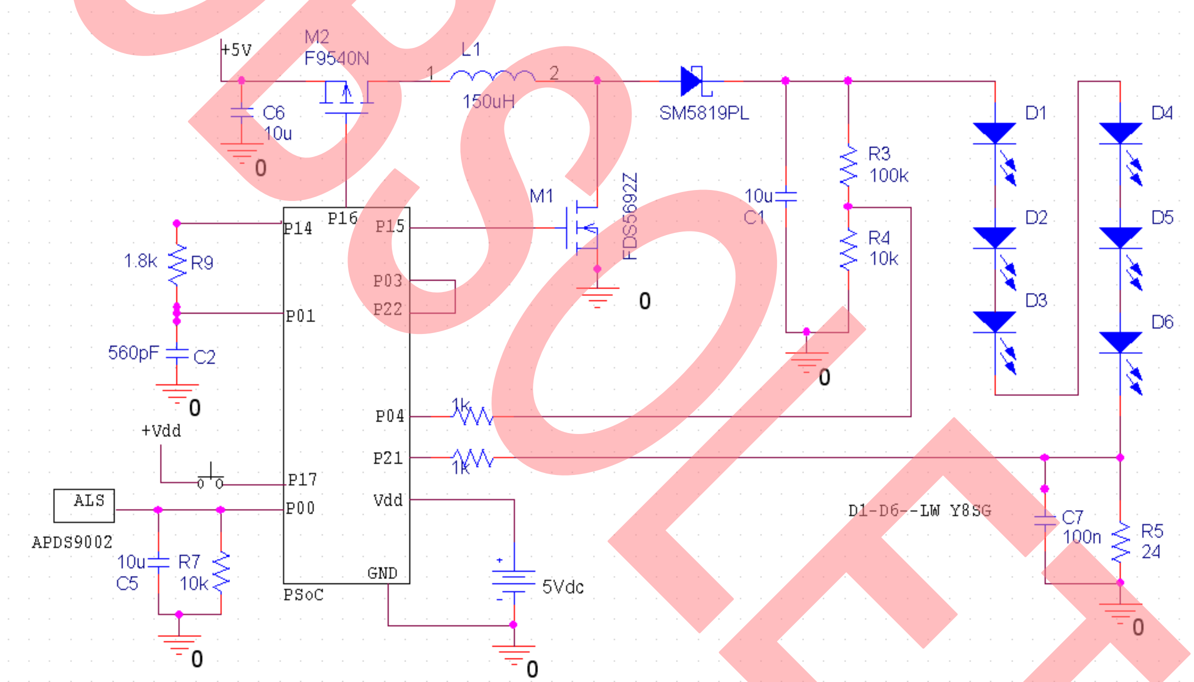
This information is useful to test the projects that accompany this application note.

Project 1: Hardware Based Backlight Control without ALS

Changing brightness: Switch is connected at Port P17. When the switch is pressed, the reference count is incremented by 1. When it reaches a maximum of 12 counts, it is reset to 3 (minimum reference).

Figure 16 shows the schematic for the project. Disconnect ALS sensor interface to device for this project.

Figure 16. Hardware Based Backlight Control with ALS



Project 2: Hardware Based Backlight Control with ALS

The relation between the backlight intensity and the ALS output is:

Backlight reference = ALS sensor value/16. Instead of this, you can create discrete brightness levels.

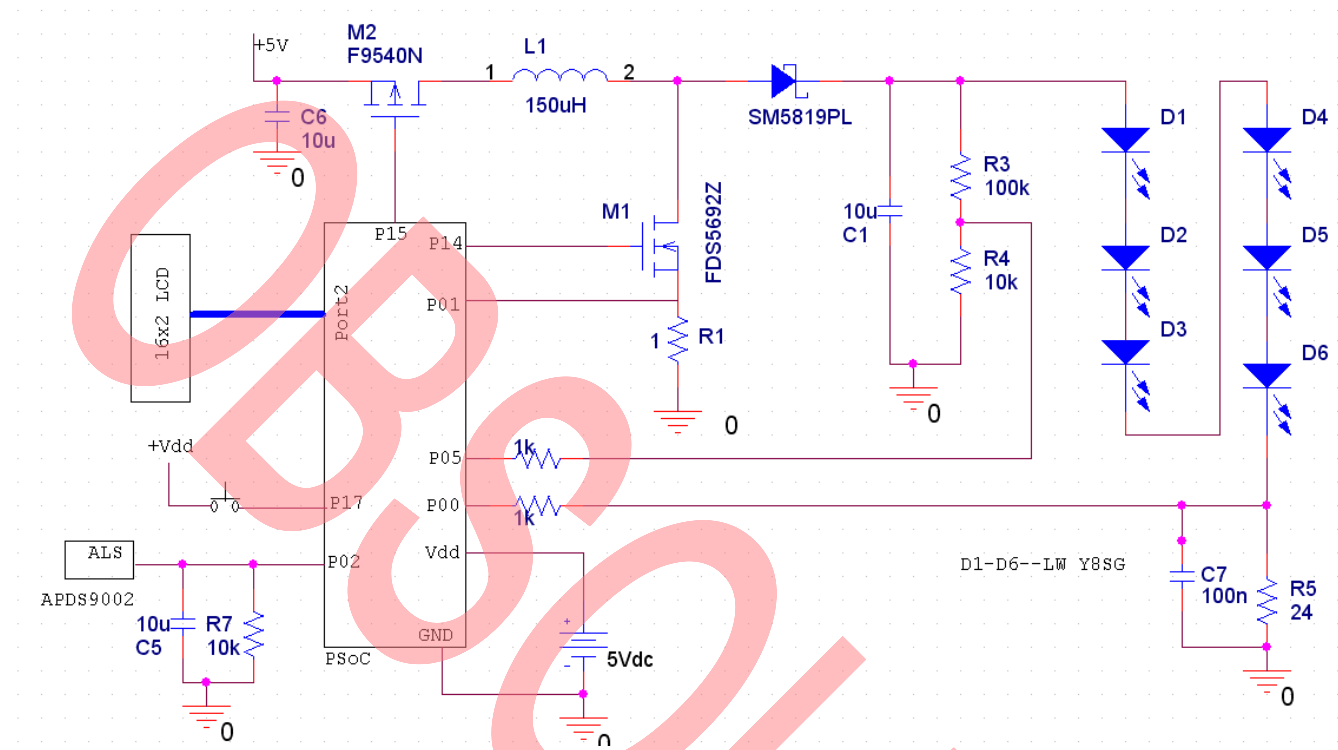
Figure 16 shows the schematic for the project.

Project 3: Firmware Based Backlight Control without ALS

Changing brightness: Switch is interfaced to the device at Port P17. When the switch is pressed, the reference count is incremented. When it reaches maximum reference value of 47 (dec), it is reset to minimum reference value of 3. The reference count, duty cycle, and ADC count for LED current are displayed on the 16x2 LCD display. Refer the LCD user module datasheet for details about the LCD.

Figure 17 shows the schematic for the project.

Figure 17. Firmware Based Backlight Control



Project 4: Firmware Based Backlight Control with ALS

The relation between the backlight intensity and the ALS output is:

Backlight reference = ALS sensor value/16.

The reference count, duty cycle, ALS value, and ADC count for LED current are displayed on the 16x2 LCD display. Refer the LCD user module datasheet for details about the LCD used.

Figure 17 shows the schematic for the project.

Table 6. BOM for Boost Network

Component	Part
MOSFET M1	FDS5692Z
MOSFET M2	F9540N
White LEDs D1-D6	LWY8SG
Inductor	Coilcraft LPS4018-154MLC
Schottky diode	SM5819PL

Appendix B

Resource Consumption in Various PSoC Parts

		Hardware Based Control with ALS				Firmware Based Control with ALS							
		CY8C24223	CY8C24794	CY8C27443	CY8C29466	CY8C21234	CY8C21223	CY8C24223	CY8C23533	CY8C24533	CY8C24794	CY8C27443	CY8C29466
Features	Open circuit	Red	Red	Green	Green	Blue	Blue	Blue	Yellow	Yellow	Blue	Green	Green
	Short circuit	Red	Red	Green	Green	Blue	Blue	Blue	Yellow	Yellow	Blue	Green	Green
	MOSFET current limit	Red	Red	Red	Red	Blue	Blue	Blue	Yellow	Yellow	Blue	Green	Green
Validated (V) /Theoretical (T)		T	T	V	T	T	T	T	T	T	T	V	T
Analog UM		CMP-1 ADCINC (8) SCBlock-2** DAC PGA	CMP-1 ADCINC (8) SCBlock-2** DAC PGA	CMP-3 ADCINC (8) SCBlock DAC PGA	CMP-3 ADCINC (8) SCBlock DAC PGA	CMP-1 ADC8	CMP-1 ADC8	CMP-1 ADCINC (8) PGA	CMP-2 SAR	CMP-2 SAR	CMP-1 ADCINC (8) PGA	CMP-3 ADCINC(8) PGA	CMP-3 ADCINC(8) PGA
Digital UM		PWM8 DigBuf	PWM8 DigBuf	PWM8 DigBuf	PWM8 DigBuf	PWM16 DigBuf*	PWM16 DigBuf*	PWM16 DigBuf*	Timer16 DigBuf	Timer16 DigBuf	PWM16 DigBuf*	Timer16 DigBuf	Timer16 DigBuf
DBB	Used	2	2	2	2	2	2	2	2	2	2	2	2
	Available	2	2	4	8	2	2	2	2	2	2	4	8
DCB	Used	1	1	1	1	2*	2*	2*	1	1	2*	2	2
	Available	2	2	4	8	2	2	2	2	2	2	4	8
ACB/ACE	Used	2	2	4	4	2	2	2	2	2	2	4	4
	Available	2	2	4	4	2	2	2	2	2	2	4	4
ASC/ASD/ASE	Used	4	4	3	3	1	1	1	0	0	1	1	1
	Available	4	4	8	8	2	2	4	2	2	4	8	8
Fixed Resources	Used	-	-	-	-	-	-	-	SAR	SAR	-	-	-
	Available	I2C	I2C,USBFS	I2C	I2C	I2C	I2C	I2C	SAR,I2C	SAR,I2C	I2C,USBFS	I2C	I2C
IO	Used	8	8	9	9	5	5	5	6	6	6	7	7
	Available	16	50	24	24	12	12	16	26	26	50	24	24
Flash	Used	-	-	1334 bytes (8.2%)	-	-	-	-	-	-	-	1668 bytes (10.4%)	-
	Available	-	-	16K	-	-	-	-	-	-	-	16k	-
RAM	Used	-	-	24 bytes (9.4%)	-	-	-	-	-	-	-	34 bytes (13.7%)	-
	Available	-	-	256	-	-	-	-	-	-	-	256	-

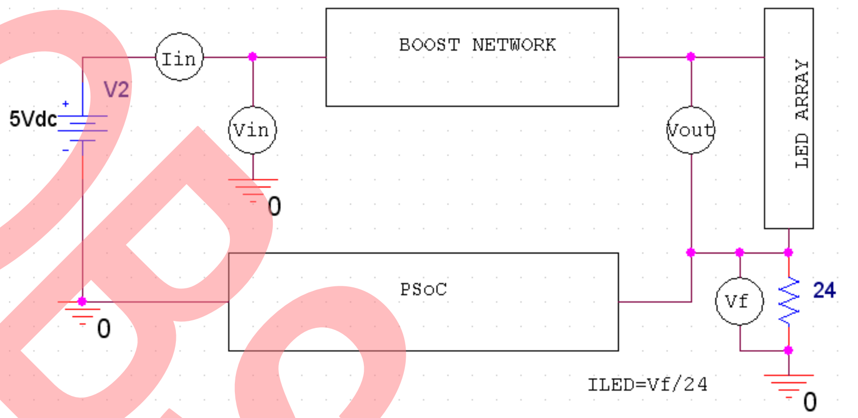
Red: No feature possible — Yellow: Any two features possible — Blue: Any one feature possible — Green: All features possible

*If MOSFET current limit is implemented, there is no need of DigBuf; PWM module must be replaced by a timer module.

**One extra SC block is used only for routing.

Appendix C

Test Setup to Measure Efficiency



$$Pout = Vout * \frac{Vf}{24}$$

$$Pin = Vin * Iin$$

$$Efficiency = \frac{Pout}{Pin}$$

Document History

Document Title: Implementing White LED Driver Using PSoC® with Ambient Light Sensor Interface – AN52581

Document Number: 001-52581

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	2682760	RJVB	04/02/09	New application note
*A	3596494	RJVB	04/23/2012	Updated in new template. Completing Sunset Review.
*B	4332340	RJVB	04/03/2014	Obsolete document. Completing Sunset Review.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Automotive	cypress.com/go/automotive
Clocks & Buffers	cypress.com/go/clocks
Interface	cypress.com/go/interface
Lighting & Power Control	cypress.com/go/powerpsoc cypress.com/go/plc
Memory	cypress.com/go/memory
Optical Navigation Sensors	cypress.com/go/ons
PSoC	cypress.com/go/psoc
Touch Sensing	cypress.com/go/touch
USB Controllers	cypress.com/go/usb
Wireless/RF	cypress.com/go/wireless

PSoC® Solutions

psoc.cypress.com/solutions

PSoC 1 | PSoC 3 | PSoC 5

Cypress Developer Community

Community | Forums | Blogs | Video | Training

Technical Support

cypress.com/go/support

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