ColorLock Optical Feedback for EZ-Color™

AN44533

Authors: Andriy Ryshtun, Vadym Grygorenko, Volodymyr Sokil, Triton Hurd
Associated Project: Yes
Associated Part Family: CY8CLED16
GET FREE SAMPLES HERE
Software Version: PSoC Designer™ 4.4
Associated Application Notes: AN15733, AN16035, AN2352

Application Note Abstract

This application note describes the implementation principles for an EZ-Color™ based RGB color mixing system with ColorLock optical feedback. PSoC® firmware and a PC GUI are also described.

Introduction

There are several implementations of color mixing systems based on EZ-Color devices. These implementations are described in application notes, AN15733 and AN16035.

In previous solutions, the color set point is based on calculations using LEDs' data sheet binning information for color and flux. This may cause inaccuracies. Color changes due to die temperature drift, forward current variation, aging, and other factors are only partially compensated by firmware. The temperature feed compensation used requires precision characterization of LED temperature and luminosity dependence. The same situation occurs with aging time compensation. There is also no way to compensate for color errors caused by individual LED color variation and ambient light interference. The resulting color error depends significantly on LED batch to batch variations.

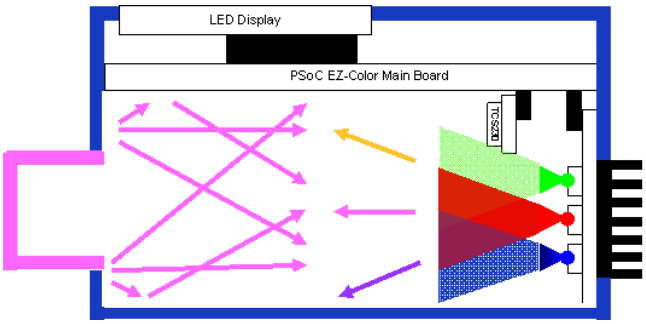
Introducing an optical feedback creates increased target color precision. It allows the color mixing system to operate independent of individual LED parameters. It also makes temperature compensation easier because chromaticity changes are compensated by optical feedback and not by temperature degradation data, which varies from LED to LED.

Calibration of the system is done easily using a reference colorimeter and PC GUI application.

Structure of Color Regulator

A color light fixture system consists of RGB LEDs, a diffuser, a mechanical enclosure, and a color sensor (Figure 1). The color sensor is placed in such a way that it receives only diffused light and avoids direct LED rays.

Figure 1. Mechanical Layout of ColorLock System



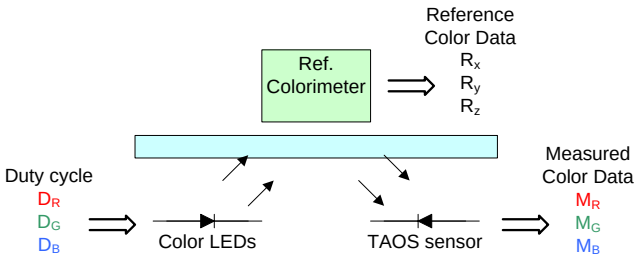
Color regulation is done using duty cycles with LED drivers. This sets the required brightness and color. Color sensor signals form a feedback loop and external host commands indicate the required set point.

The main challenge is that the color spaces of the color sensor, LEDs, and host controller are completely different. The host controller uses CIE1931 XYZ tri-stimulus values. The color sensor measures three RGB values in units which are not related to any standard color coordinate. Also, if the luminosity of any LED changes, it affects all three sensor output values. This means that it is impossible to build three independent regulators for R, G, and B LEDs based on corresponding sensor channels.

Mathematical Description of Color Mixing System with Feedback

The simplified block diagram (Figure 2) illustrates the relationship between all values describing the color regulator.

Figure 2. Block Diagram of Color Regulator Components



There are three backlight LEDs: red, green, and blue. The luminance of these LEDs is defined by the corresponding driver duty cycles, D_R, D_G, and D_B. The duty cycle range is assumed to be between 0 and 100%. The real duty cycle code range depends on the used PWM (or in this case PrISM[1]) resolution and is calculated by the output vector D scaling.

The resulting color of the diffuser is measured by the TAOS TCS230 color sensor. This sensor output data is three numbers, corresponding to red, green, and blue channels, M_R, M_G, and M_B. To calibrate the whole system, an external reference colorimeter is used. The reference colorimeter output is standard CIE XYZ color coordinates, R_X, R_Y, and R_Z. The reference colorimeter XYZ values and target color XYZ coordinates are equal (proportional) if the color regulation system works properly.

The following equations are solved to build the color regulator:

- Convert the XYZ set point (target color) R into target color sensor output M.
- Convert color sensor output M into LED driver duty cycle D.
- If the measured color data changes due to environmental changes or other reasons, then update the LED driver duty cycle D to keep the color sensor output stable.

Due to the linear nature of LEDs' luminosity vs. driving duty cycle and sensor outputs vs. input light, there must be a linear matrix transformation that translates $\mathbf{D} \rightarrow \mathbf{M}$, $\mathbf{M} \rightarrow \mathbf{R}$ and vice versa.

$$M = S \cdot D \quad \text{Equation 1}$$

$$R = C \cdot M \quad \text{Equation 2}$$

or, using Equation 1,

$$R = C.S \cdot D \quad \text{Equation 3}$$

where S is a square 3x3 matrix that translates duty cycles into sensor signals. C is a square 3x3 matrix that translates sensor signals into CIE XYZ color coordinates.

Matrices Estimation by Calibration

Unfortunately, there is no way to estimate numerical values of these matrices. Therefore, a dedicated one time calibration procedure is required to estimate these values. This is a simple procedure—turn on red LED to 100% ($D_R = 1$) and turn off other LEDs ($D_G = 0$; $D_B = 0$). Here,

$$\begin{pmatrix} M_R^R \\ M_G^R \\ M_B^R \end{pmatrix} = \begin{pmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} s_{11} \\ s_{21} \\ s_{31} \end{pmatrix}$$

In other words, the first column of the matrix S is the measurement result when only the red LED is turned on ($\mathbf{M}^R$). The second column is the measurement result when only the green LED is turned on ($\mathbf{M}^G$) and the third column is the result when only the blue LED is turned on ($\mathbf{M}^B$):

$$\mathbf{S} = \begin{pmatrix} M_R^R & M_R^G & M_R^B \\ M_G^R & M_G^G & M_G^B \\ M_B^R & M_B^G & M_B^B \end{pmatrix} \quad \text{Equation 4}$$

To estimate the matrix C that translates sensor signals into CIE XYZ color coordinates (Equation 2), use Equation 3 at the first step to get the product $\mathbf{C} \cdot \mathbf{S}$. Later calculate the matrix C. If

$$T = C \cdot S \quad \text{Equation 5}$$

then

$$R = T \cdot D$$

This is very close to Equation 1. The same technique is used in matrix S estimation—turning on all LEDs alternatively and storing colorimeter results:

$$\mathbf{T} = \begin{pmatrix} R_X^R & R_X^G & R_X^B \\ R_Y^R & R_Y^G & R_Y^B \\ R_Z^R & R_Z^G & R_Z^B \end{pmatrix} \quad \text{Equation 6}$$

Using Equation 5

$$C = T \cdot S^{-1} \quad \text{Equation 7}$$

Note

1. PrISM is formally known as SSDM, which is Cypress' proprietary spread spectrum dimming method. PrISM and SSDM are used interchangeably in this document.

If a reference colorimeter is not available and color precision requirements are not strong, it is possible to use approximated values of matrix C. An example of C^{-1} values is depicted by Figure 12. But S matrix must be estimated for each design because its measurement does not require colorimeter usage.

The algorithm for matrices estimation is:

1. Turn on only the red LED. Store the TAOS sensor measurement results in the first column of matrix S (Equation 4) and the colorimeter results in the first column of matrix T (Equation 6).
2. Turn on only the green LED. Store the TAOS sensor measurement results in the second column of matrix S (Equation 4) and the colorimeter results in the second column of matrix T (Equation 6).
3. Turn on only the blue LED. Store the TAOS sensor measurement results in the third column of matrix S (Equation 4) and the colorimeter results store in the third column of matrix T (Equation 6).
4. Calculate matrix C using Equation 7.

Regulator

The ColorLock system is assumed to be without inertia. Therefore, the full PID regulator usage is not mandatory. The simplest astatic regulator with only an integral channel is used here.

For practical regulator realization, the continuous time model is replaced by a discrete time model. A commonly used discrete time form of PID regulator equation is:

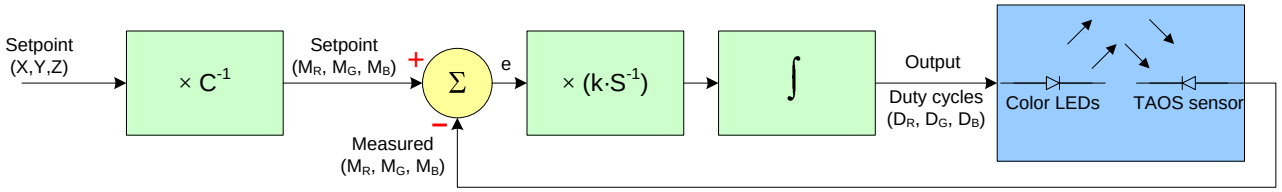
$$\begin{aligned} Output_{n+1} &= Output_n + P_n + I_n + Dif_n; \\ P_n &= K_P\ \ e_n - e_{n-1}\ ; \\ I_n &= K_I e_n; \\ Dif_n &= K_D\ \ e_n - 2e_{n-1} + e_{n-2}\ . \end{aligned} \qquad \text{Equation 8}$$

P_k, I_k, Dif_k are proportional, integral, and derivative channels values, e_k is regulator error—the difference between the set point and measured signal. Here, only the integral channel is used, which simplifies the final formula. For the regulator depicted in Figure 3 the formula is:

$$\mathbf{D}_{n+1} = \mathbf{D}_n + k \cdot \mathbf{S}^{-1} \cdot \mathbf{e}_n \qquad \text{Equation 9}$$

To avoid overshooting, the gain for regulator is $k = 1/4$.

Figure 3. Block Diagram of Regulator



Vector Normalization

RGB LEDs used for color mixing systems have different maximum luminosity. The absolute maximum luminosity depends on the target color. This means that for some set points (X, Y, Z), the required duty cycles can exceed the 100% limit. To avoid this saturation, the input information about the required color is divided into chromaticity and relative brightness values. Real brightness is recalculated for each set point and may change depending on the required color and LEDs used. The absolute brightness changes due to changes in LED temperature are corrected during runtime.

The target color is assumed to be set in xyY format, where xy corresponds to the CIE chromaticity coordinates and Y is relative luminosity; $0 \le Y \le 1$. To normalize the duty cycle vector D, first estimate the maximum luminosity Y_{max} that corresponds to saturation of any of the LEDs. Using formulas to transform $xyY \rightarrow XYZ$ and assuming $Y = 1$,

$$\begin{aligned} X &= Y\frac{x}{y}; \\ Y &= Y; \\ Z &= Y\frac{1-x-y}{y}. \end{aligned} \qquad \text{Equation 10}$$

If $Y = 1$, then

$$\begin{aligned} X_0 &= \frac{x}{y}; \\ Y_0 &= 1; \\ Z_0 &= \frac{1-x-y}{y}. \end{aligned} \qquad \text{Equation 11}$$

From Equation 3,

$$\mathbf{D}_0 = \mathbf{S}^{-1}\mathbf{C}^{-1} \cdot \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \end{pmatrix} \qquad \text{Equation 12}$$

The vector of duty cycles $\mathbf{D}_0$ corresponding to $Y = 1$ contains values above 100%. Because this cannot be implemented, a scale factor $Y_{\max}$ is introduced for normalization of duty cycle vector components to 100%:

$$Y_{\max} = 1/\max\ D_R, D_G, D_B \qquad \text{Equation 13}$$

For the given chromaticity xy and luminosity $Y_{\max}$, the resulting duty cycle vector contains at least one component with 100% value. Using this value, the final set point for the regulator is calculated.

$$\mathbf{M}_{set} = Y \cdot Y_{\max} \cdot \mathbf{C}^{-1} \cdot \begin{pmatrix} X_0 \\ 1 \\ Z_0 \end{pmatrix} \qquad \text{Equation 14}$$

This formula allows easy set point recalculation on target relative luminosity Y changes or on maximum luminosity $Y_{\max}$ changes due to LED's heating. If a duty cycle exceeds the 100% limit, $Y_{\max}$ values are recalculated during normal regulator operation.

$$Y_{\max}(new) = Y_{\max}(old)\frac{1}{D_{\max}} \qquad \text{Equation 15}$$

where, $D_{\max}$ is the duty cycle value that exceeds 100% limit.

The vector normalization algorithm is:

1. Target color is received in xyY format, where xy corresponds to CIE chromaticity coordinates and Y is the relative luminosity; $0 \le Y \le 1$.
2. Using Equation 11 to Equation 13 calculate maximum luminosity $Y_{\max}$.
3. Calculate the set point according to Equation 14.
4. If the required relative luminosity Y changes, recalculate the set point using Equation 14.
5. If LED temperature changes cause a correction of $Y_{\max}$, recalculate the set point using Equation 14.
6. If a duty cycle exceeds 100% due to a correction of $Y_{\max}$, recalculate the set point using Equation 14.

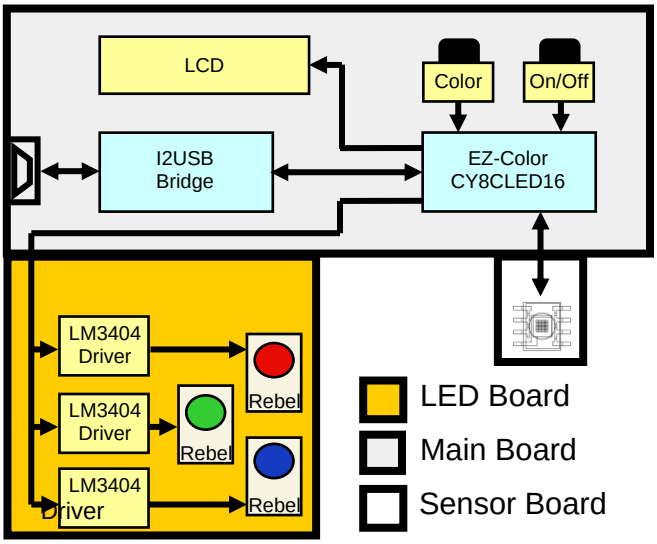
Hardware Architecture

The main tasks of the hardware are:

- Form driving signals for three LED drivers with 9-bit resolution.
- Measure the frequency of the color sensor output signal.
- Provide consistently accurate color output despite external condition changes.

A simplified block diagram of the hardware is shown in Figure 4. There are three boards in the system. The main board contains the EZ-Color device, push buttons, LCD, and the I²C™ to USB bridge. The LED board contains the LEDs, drivers, and temperature sensor. The sensor board contains only the sensor. The system is developed to be modular so different LEDs and sensors can be used without a complete redesign.

Figure 4: Hardware Block Diagram



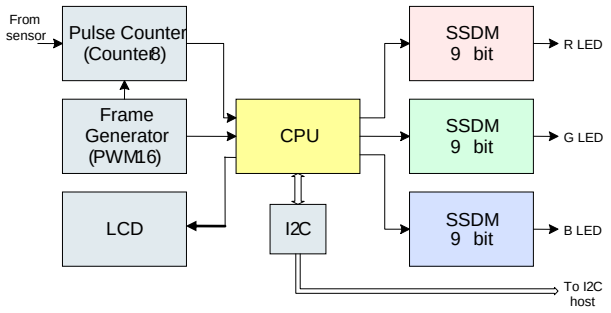
A simplified block diagram of the EZ-Color configuration is depicted in Figure 5. The schematics are illustrated in Appendix A.

The CPU measures the frequency from the color sensor signals using a pulse counter and frame generator. Measured values are compared to the set point and the difference obtained is used to correct the LED's brightness by a corresponding change of the LED drivers' duty cycles. LED driving signals are formed by Stochastic Signal Density Modulation (SSDM) user modules. Duty cycles (signal density) are calculated by the CPU and written into the hardware modules. SSDM output signals drive low power LEDs directly or via dedicated LED drivers such as LM3402, for high power LEDs.

The LCD displays the target and the measured xy coordinates. The “On/Off” push button turns the LEDs on and off; the “Color” push button cycles through different colors.

The EZ-Color works as standalone device. It does not require any interaction with the external host to operate. However, it contains an I^2C interface for communication with external host. That host sends the required color, brightness, and calibration values and reads various data for debug purposes.

Figure 5: EZ-Color Configuration Block Diagram

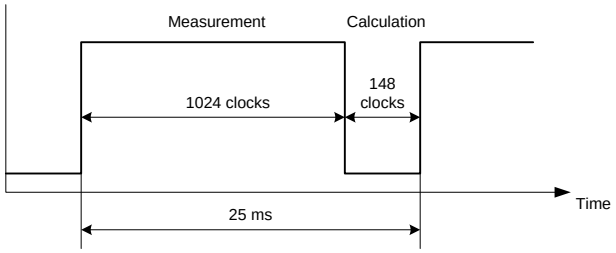


SSDM user modules form LED driving signals. There are three LEDs; therefore, six digital blocks are required for SSDMs. Additional three digital blocks are used to measure frequency of color sensor output signal. Thus, nine digital blocks are required to build the ColorLock system. That requirement defines CY8CLED16 device choice as the system’s hardware platform.

Most of the hardware blocks work in standard mode. There is only a peculiarity in the input frequency measurement. Frequency measurement is based on input pulse counting during a fixed period. This period must be synchronized to the period of SSDM. Such synchronization is mandatory because the input frequency is not stable and changes according to the SSDM output sequence. If the measurement time is equal to the SSDM period, then the sequence during each measurement is the same and the result is stable. To achieve synchronization, the PWM16 user module is clocked by the same source as the SSDM user module and the pulse width is set equal to the SSDM period (2^{10}). The Compare Out signal is connected to the Enable input of Counter8 (Figure 6). The interrupt is raised on this signal’s High→Low transitions, which allows reading data from Counter8 and data proceedings. When the Enable signal is low, the firmware stores measured data, reinitializes Counter8, and configures the TAOS sensor channel to measure the next color value (R-G-B).

To increase the capacity of the counter a dedicated variable is increased on Counter8 overflow using its Terminal Count interrupt. This allows a frequency measurement with a resolution up to 16 bits.

Figure 6. Counter8 Enable Signal



The I^2C temperature sensor is not used in this project, but is on the board for future use, if needed.

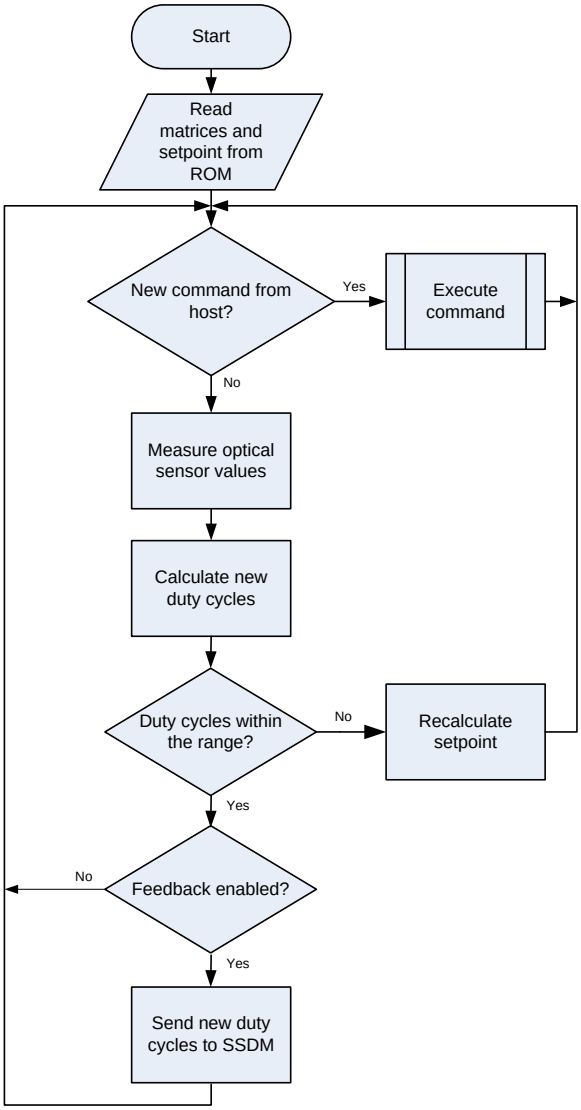
Firmware

All main operations for color regulation are performed by the EZ-Color firmware. These operations are:

- Set point calculation
- Color sensor signal measurement
- Duty cycle calculation for LED drivers
- Communication with the host
- Push button and LCD control

All communication with host is performed through the I^2C interface. The I^2C master device, which is an USB to I^2C bridge, sends predefined commands to set the target color, perform calibration, and store calibration results in an internal EEPROM. The I^2C master device also reads current values of the duty cycles and color sensor measurement results for debug purposes.

Figure 7. PSoC Firmware Flowchart



I²C Interface Description

All communication between EZ-Color and the external host (PC) is performed through an I²C interface. EZ-Color contains the EzI2Cs user module with address 0x28 (8-bit I²C address).

Table 1. I²C Interface Description

Addr	Data Type	Mode	Name	Description
00	BYTE	w	bCommand	Command byte. Set different operation modes
01	WORD	w	wx	x value
03	WORD	w	wy	y value
05	WORD	w	wY	Y value (brightness)
07	BYTE	w	bPointer	Matrix write pointer
08	float	r/w	fK0	S\C_1_Matrix[wPointer][0]
12	float	r/w	fK1	S\C_1_Matrix[wPointer][1]
16	float	r/w	fK2	S\C_1_Matrix[wPointer][2]
20	BYTE	w	bWriteFlag	Write complete flag
21	WORD	r	wSensorR	RED sensor signal
23	WORD	r	wSensorG	GREEN sensor signal
25	WORD	r	wSensorB	BLUE sensor signal
27	WORD	r	wLED_R	Current Red LED SSDM value
29	WORD	r	wLED_G	Current Green LED SSDM value
31	WORD	r	wLED_B	Current Blue LED SSDM value
33	WORD	r	wMr	Set point Red
35	WORD	r	wMg	Set point Green
37	WORD	r	wMb	Set point Blue
39	WORD	r	X	Measured Color x value
41	WORD	r	Y	Measured Color y value
43	WORD	r	U	Measured Color u value
45	WORD	r	V	Measured Color v value

Byte **bCommand** defines the operation mode and other data destinations. Allowed values are:

- 0—Color feedback is enabled. Accept wx, wy, wY data.
- 1—Calibration mode. Color feedback is disabled.
  wx = Red LED SSDM value;
  wy = Green LED SSDM value;
  wY = Blue LED SSDM value.
- 2—Write S_1 Matrix into EEPROM. Color feedback is disabled.
  S_1_Matrix[bPointer][0] = fK0;
  S_1_Matrix[bPointer][1] = fK1;
  S_1_Matrix[bPointer][2] = fK2;

- 4—Write C_1 Matrix into EEPROM mode. Color feedback is disabled.
  C_1_Matrix[bPointer][0] = fK0;
  C_1_Matrix[bPointer][1] = fK1;
  C_1_Matrix[bPointer][2] = fK2;

- 8—Write C Matrix into EEPROM mode. Color feedback is disabled.
  C_Matrix[bPointer][0] = fK0;
  C_Matrix[bPointer][1] = fK1;
  C_Matrix[bPointer][2] = fK2;

Word values **wx**, **wy**, and **wY** correspond to the xyY target color. For convenience, these values are scaled to the 0..512 integer range.

Byte **bPointer** is used to set calibration matrices $\mathbf{C}$, $\mathbf{C}^{-1}$ and $\mathbf{S}^{-1}$; it indicates the row number.

Byte **bWriteFlag** is used as a data write complete sentinel. It must be equal to 1.

An example of an I^2C sequence for sending xyY data is shown in Table 2.

Table 2. I^2C Command Example

Write I^2C Command	Command byte	xyY values	Not Used	Write Complete Flag
S 28 0	0	03 AA 03 AA 03 AA 0	0 0 0 0 0 0 0 0 0 0 0 0	1

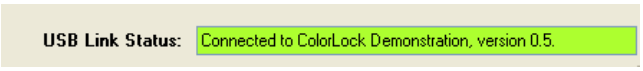
PC Application

The main purpose of the PC GUI application is to demonstrate the ColorLock system features. It allows selecting a target color and brightness, while displaying the real time sensor values, duty cycles, and adjustment inaccuracy. The application also makes it possible to perform interactive system calibration using a reference colorimeter.

General Description

When the application is launched, the "USB Link Status" field in the bottom right of the window must be green and read "Connected to ColorLock Demonstration" (Figure 8).

Figure 8. USB Link Status Field.



The main window of the application is shown in Figure 10. There are two chart panels in the top portion of the main window. The left chart, *Current CIE Sensor Coordinates,* presents actual CIE x, y (or u, v) values measured by the TAOS sensor. The right chart, *ColorLock Accuracy,* displays the current absolute difference between the sensor signals and the set point. These charts specifically allow observing the ColorLock transient on set point or ambient light changes.

The bottom left of the window contains a color picker and brightness track bar. The color picker looks similar to the xy CIE1931 or u'v' CIE1976 color space depending on the mode. It allows the user to select the color coordinates of the target color. The relative brightness is selected using the track bar. The corresponding CIE coordinates and color sample are displayed in the edit boxes at the bottom part of the window. These edit boxes are used to set the target CIE color coordinates manually.

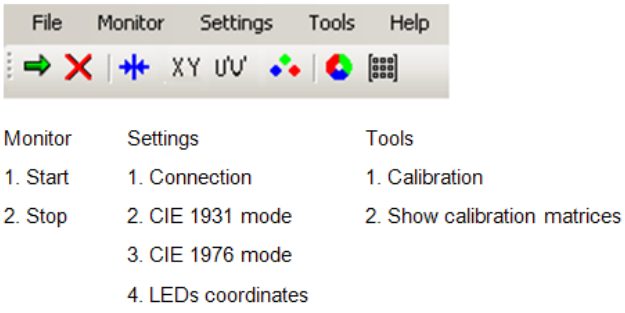
The bottom right of the main window contains three track bars for PrISM duty cycle (signal density) values. These track bars have two functions depending on the *Disable Feedback* checkbox state. By default this checkbox is unchecked, which means that the system works in normal mode with optical feedback. In this state the track bars display actual values of the PrISM output signals. If *Disable feedback* is checked it means that the user can directly send PrISM values using corresponding track bars. *SSDM capacity* combo box gives the possibility to define the size of the SSDMs used in the project—9 or 10 bits.

Another function of the GUI is system calibration using a reference colorimeter. The simple wizard (Figure 11) allows input of XYZ data from the colorimeter when red, green, and blue LEDs are turned on. Calibration matrices are calculated by the application and are sent to the device using commands 2, 4, and 8 (see Table 1). The values of the matrices are inspected in the Calibration Matrices window (Figure 12). The calibration process gives the possibility to obtain precise color coordinates of the LEDs (Figure 13).

Menu and Toolbar Description

The main menu and toolbar structure is shown in Figure 9.

Figure 9. Main Menu and Toolbar Structure



File > Exit menu item stops the monitor if running, and closes the application.

Monitor > Start menu item () starts the application monitor mode.

Monitor > Stop menu item () stops the monitor.

Settings > USB-I2C Connection menu item () shows the *Connection Center* dialog box.

Settings > Color Space > CIE1931 menu item () switches the application into xy CIE1931 color space mode. All color coordinates and chart values in the application are visualized according to this color space.

Settings > Color Space > CIE1976 menu item (u'v') switches application into u'v' CIE1976 color space mode.

Settings > LEDs Color Coordinates menu item ( ) shows the *LEDs' Color Coordinates* dialog (Figure 13). Independent from active color space settings, LED coordinates are visualized and accepted in xy CIE1931 color space coordinates.

Tools > Calibration menu item ( ) launches the Calibration Wizard.

Tools > Show Calibration Matrices menu item ( ) shows the *Calibration Matrices* window (Figure 12).

Figure 10. Main Window: Set Color Process

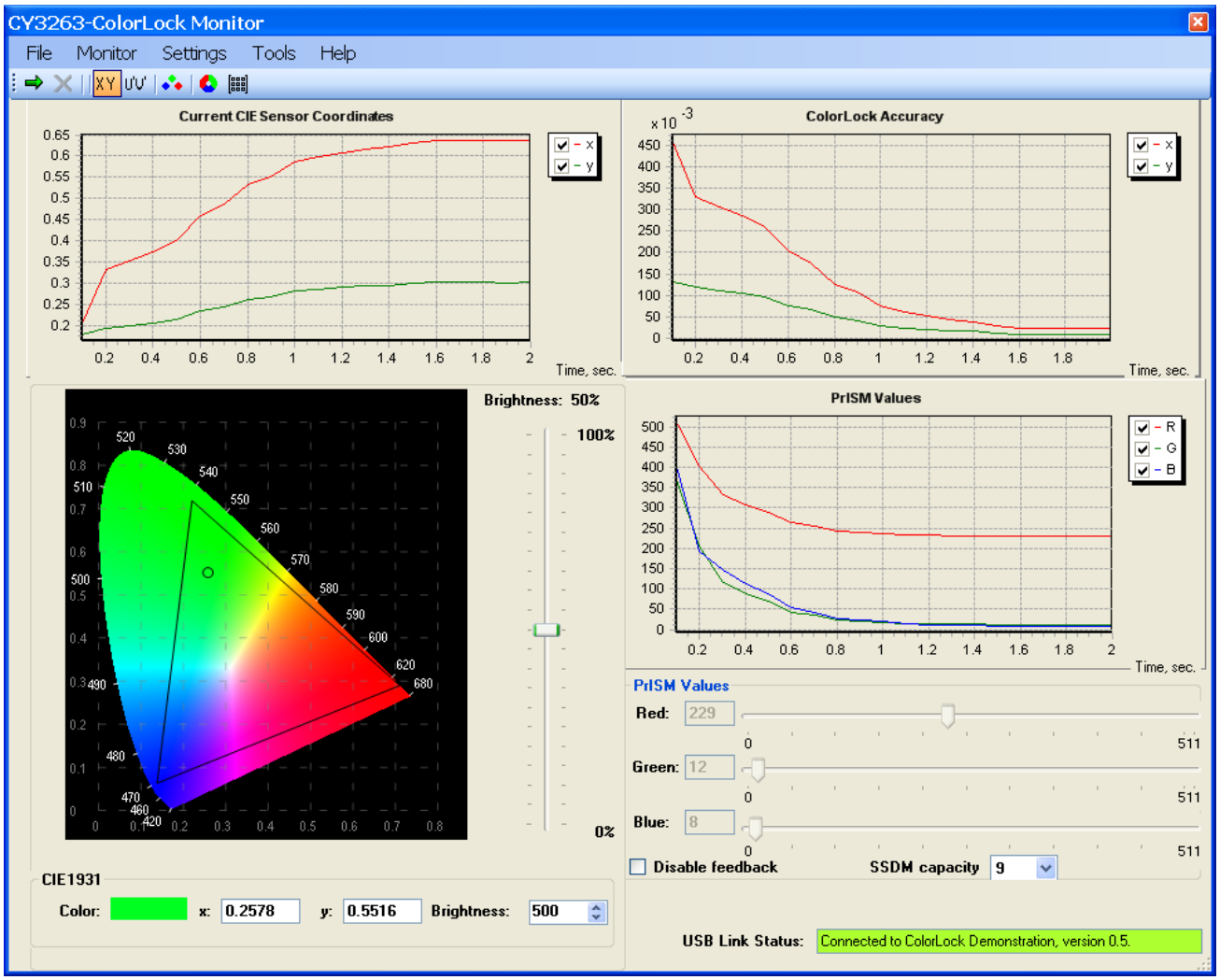


Figure 11. Calibration Data Dialog

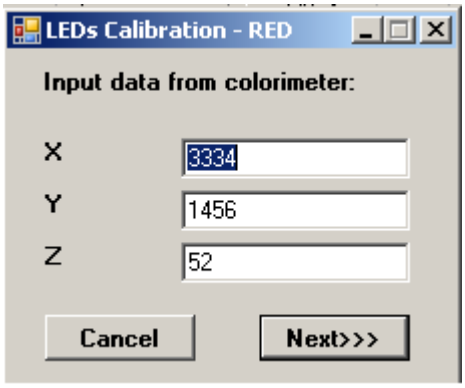


Figure 12. Calibration Matrices Form

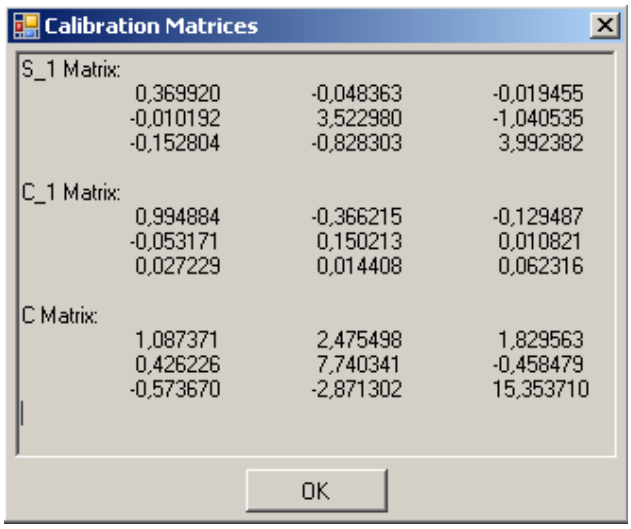
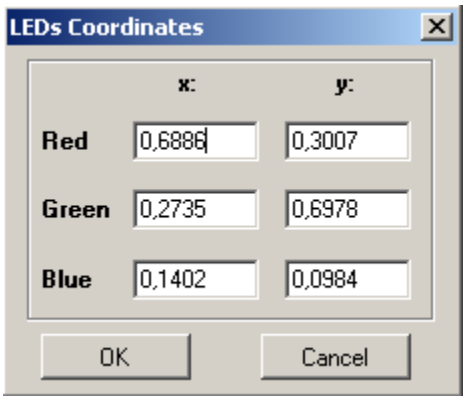


Figure 13. LED Color Coordinates



Appendix A

Schematics

Figure 14. ColorLock Rebel Daughter Card

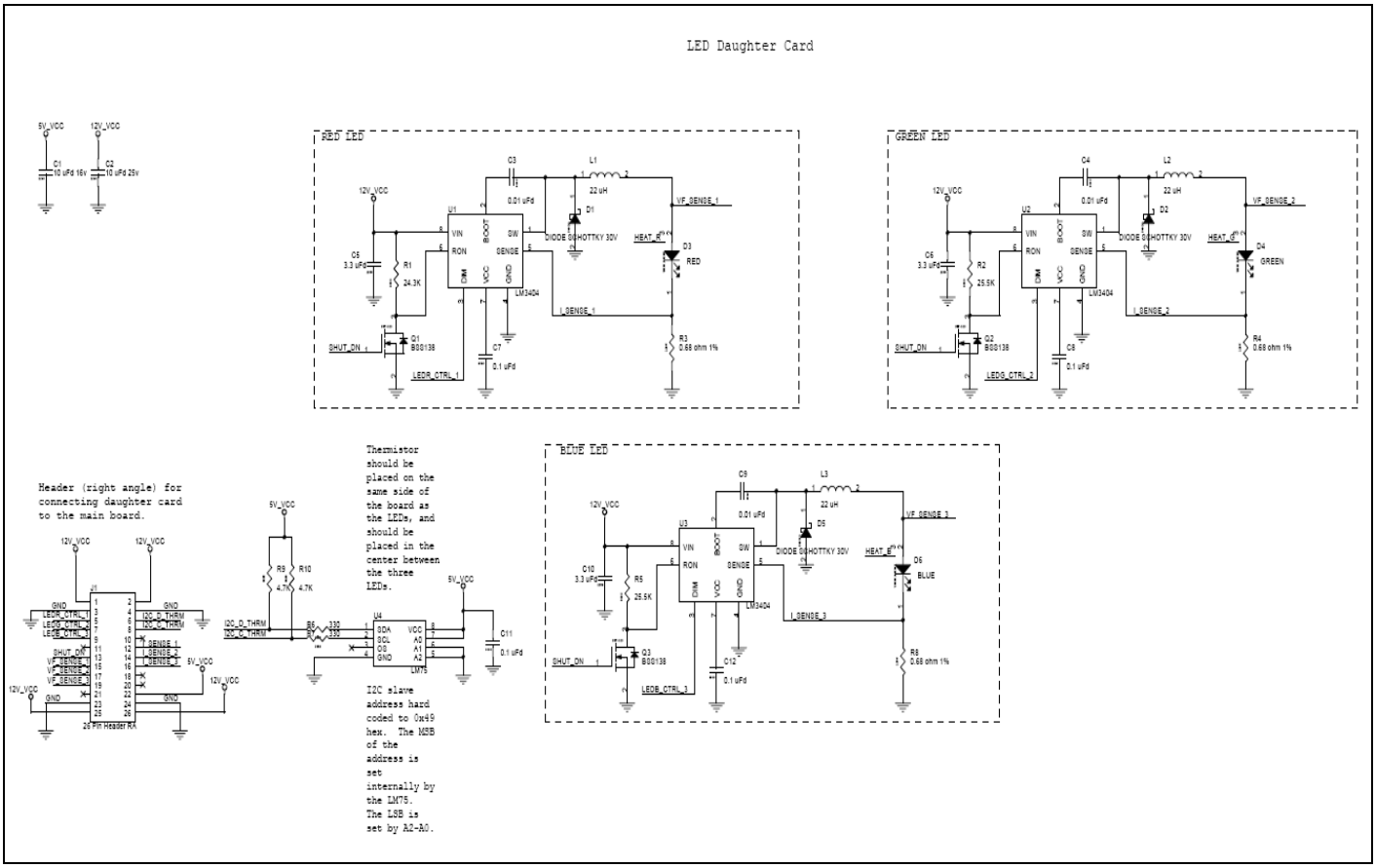


Figure 15. ColorLock Main Board

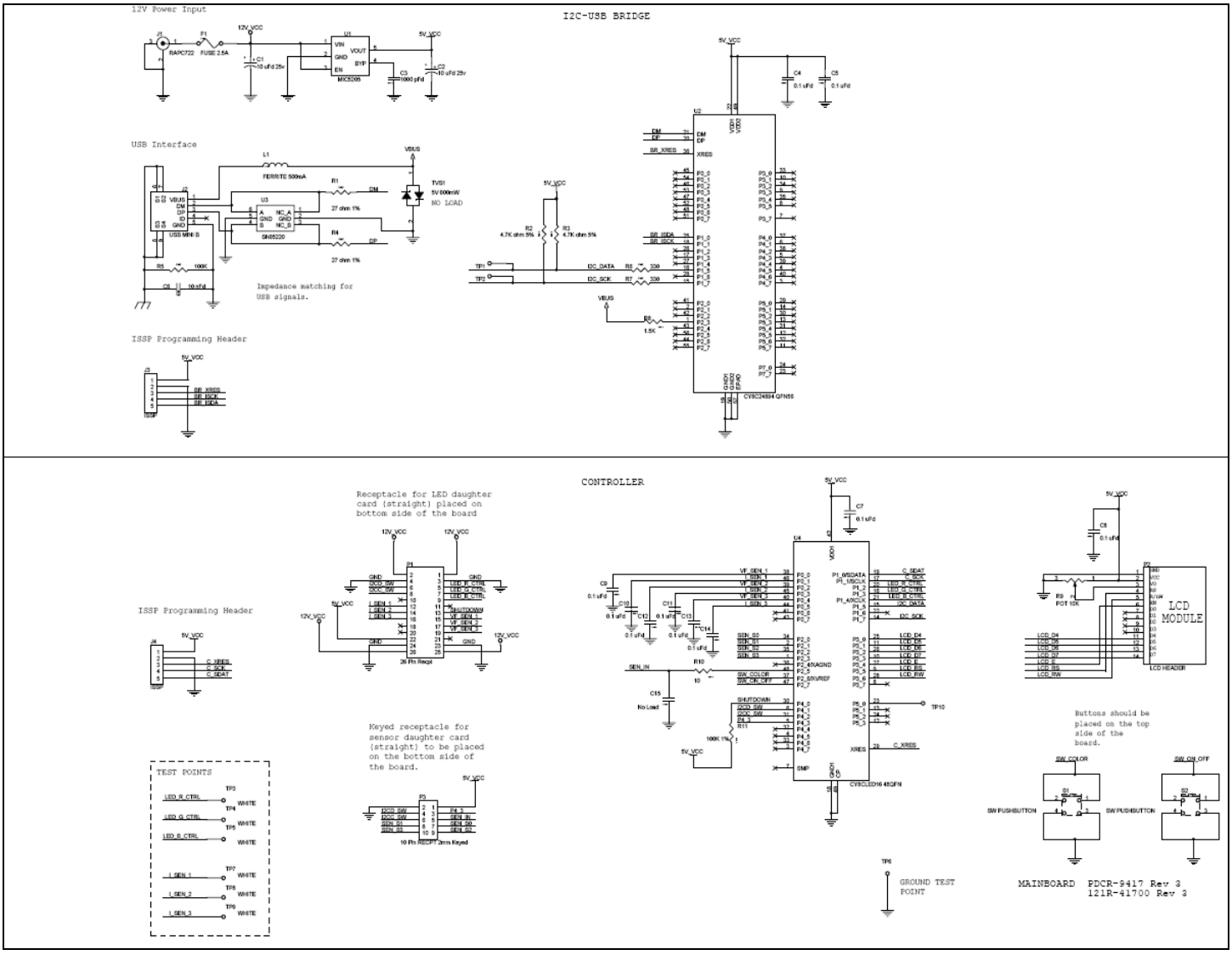
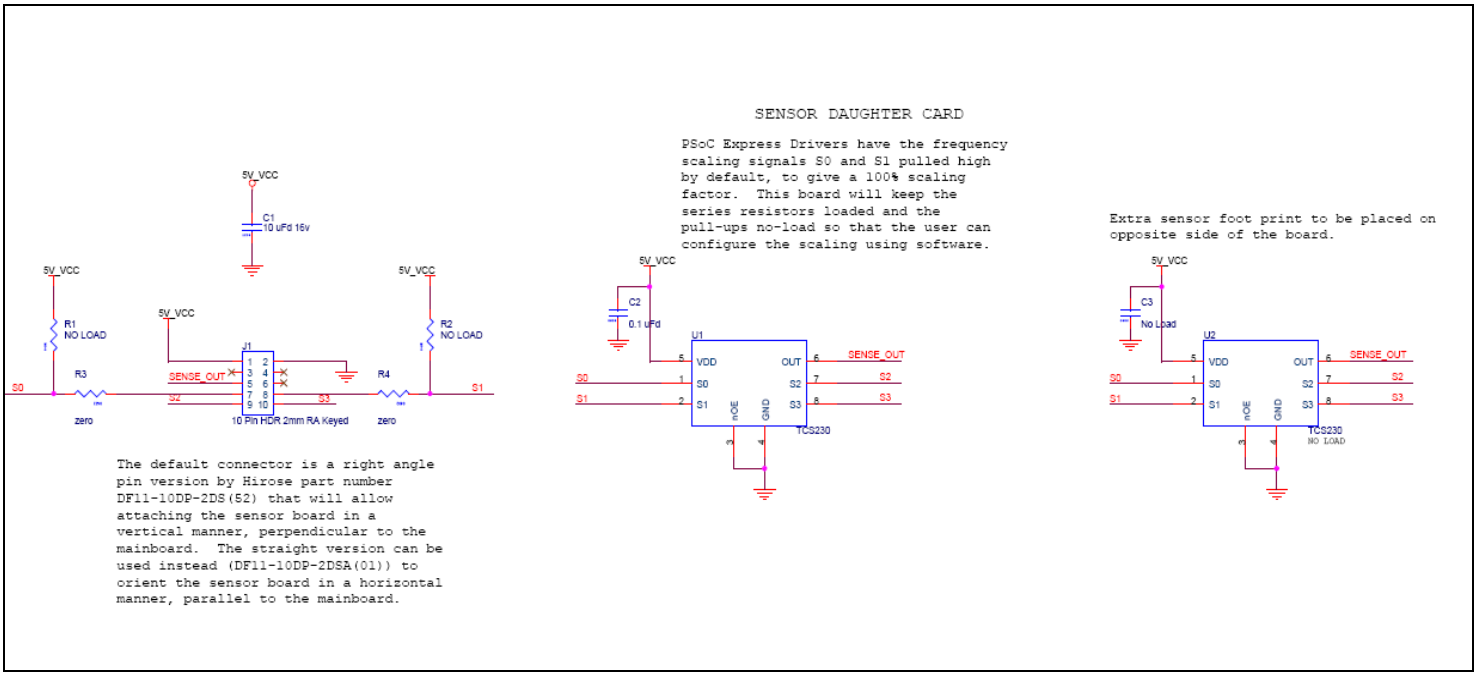


Figure 16. ColorLock TAOS Daughter Card



About the Author

Name:	Andriy Ryshtun
Title:	Ukraine Solution Center Applications Engineer
Background:	More than two years in USC. Experience in analog electronics, CapSense, and PCB design.
Contact:	Andriy.Ryshtun@cypressua.com

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