AN52491

Implementing Ambient Light Sensing Using PSoC® 1

Author: Jaya Kathuria

Associated Part Family: CY8C23X33, CY8C24x23, CY8C24X33, CY8C21X23 CY8C21X34,CY8C24X94, CY8C27X43,CY8C29X66

Related Application Notes: AN75320, AN2397

To get the latest version of this application note, or the associated project file, please visit http://www.cypress.com/go/AN52491.

AN52491 describes how to implement ambient light sensing using PSoC® 1. Two example projects interfacing an external analog ambient light sensor are also presented.

Contents

1 Introduction 1
2 What are Ambient Light Sensors? 2
3 Features of Ambient Light Sensors 2
4 Applications 3
5 LX1972A Description 3
6 Example Project: ALS_ADC Example 4
6.1 Device Configuration 4
6.2 Hardware Requisites 8
6.3 Test Procedure 8
6.4 Expected Results 9
7 Example Project: ALS_Comparator Example 11
7.1 Device Configuration 11
7.2 Hardware Requisites 12
7.3 Test Procedure 12
7.4 Expected Results 13
8 Summary 13
9 Appendix A 14
9.1 Power Calculations 14
10 Document History 15
Worldwide Sales and Design Support 16
Products 16
PSoC® Solutions 16
Cypress Developer Community 16
Technical Support 16

1 Introduction

Ambient light sensors are included in mobile phones, laptops, and handheld devices. They sense the environment lighting conditions and adjust the screen's backlight to comfortable levels. Studies have shown that backlighting is required only for 40 percent of the usage time. Therefore, an automatic adjustment (auto dimming) of the backlight offers considerable power savings.

This application note describes the firmware and hardware to interface ambient light sensors, and the signal processing of analog signals using the PSoC device. It also discusses the advantages of using PSoC in this solution.

Note This application note discusses the implementation with PSoC® 1 devices and any reference to PSoC in the document will refer to PSoC 1. To learn more about PSoC 1, refer to AN75320 - Getting Started with PSoC® 1.

2 What are Ambient Light Sensors?

Ambient light sensors are photo detectors designed to perceive brightness in the same way as human eyes. They are used where the settings of a system must be adjusted to the ambient light conditions as perceived by humans. Two common photo detectors used in ambient light sensing are phototransistor and photodiode. Both phototransistors and photodiodes generate an output signal in response to a light input from the electromagnetic spectrum. This is shown in Figure 1. Table 1 shows their comparison.

3 Features of Ambient Light Sensors

Ambient light sensors must have the following features:

- Provide an output value that reflects the frequency sensitivity (V-lambda characteristics) of the human eye.
- High accuracy over a wide illumination range
- Low temperature coefficient
- Compact, surface-mountable package, particularly for handheld applications

Figure 1. Phototransistor, Photodiode, and Visible Light Region of the Electromagnetic Spectrum

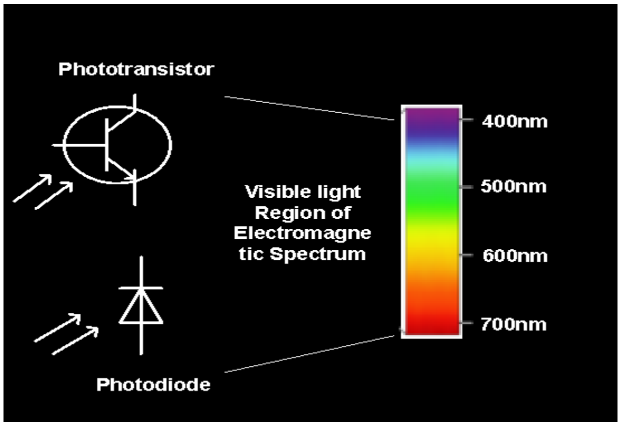


Table 1. Phototransistor versus Photodiode Comparison

Phototransistor	Photodiode
Small devices with good performance	High performance but at a larger size
Slower response time	Faster response time
Higher sensitivity (electrical output per optical input)	Lower sensitivity (electrical output per optical input)
Lower stability with temperature	High stability with temperature
Suitable for mobile applications	Suitable for automotive applications

4 Applications

- Detection of ambient light to control display backlighting in:
  - Mobile devices - mobile phones, personal digital assistants (PDA), personal media players.
  - Computing devices - TFT LCD monitor for note book computer.
  - Consumer devices – TFT LCD TV, Plasma TV, video camera, digital still camera.
- Automatic residential and commercial lighting management.
- Automatic dimming of instruments in automobiles to ensure reliable visibility.
- Headlamp control in cars to improve road safety by automatically turning on the lights in twilight or when entering a tunnel.

5 LX1972A Description

The LX1972A sensor from Microsemi is used to demonstrate the example projects accompanying this application note. The LX1972A is a low cost silicon light sensor with a spectral response that closely emulates the human eye. LX1972A provides a linear, accurate, and very repeatable current transfer function. High gain current mirrors on the chip multiply the PIN diode photo-current to a sensitivity level that can be voltage scaled with a standard value external resistor.

Table 2 summarizes the important parameters of LX1972A.

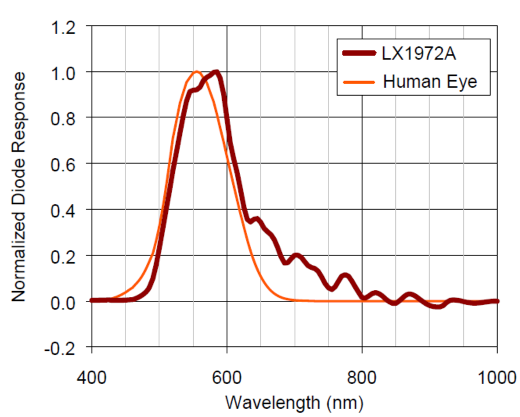
Table 2. LX1972A Specifications

Parameter	Value
Minimum Operational Voltage	Input irradiance, Ev = 100 lx, $V_{DD} – V_{SS}$ = 2 V (Worst case) Input irradiance, Ev = 1000 lx, $V_{DD} – V_{SS}$ = 2.7 V (Worst case) Input irradiance, Ev = 2000 lx, $V_{DD} – V_{SS}$ = 3 V (Worst case)
Output (Photocurrent)	Typical 235 µA at V_{CC}=5.0 V, Ev =1000 lx (Fluorescent light is used as light source)
Dark Current	50 nA at V_{CC}=5.0 V, Ev=0 lx, T_A = 25°C
Wavelength of Maximum Sensitivity (Range)	580 nm (360 to 650 nm)

Figure 2 shows its spectral response versus wavelength.

The LX1972A is ideal for applications in which the measurement of ambient light is used to control display backlighting such as mobile phones and PDAs.

Figure 2. Relative Spectral Response versus Wavelength



6 Example Project: ALS_ADC Example

The example project demonstrates how easy and effective it is to interface and process the signal from ALS sensors with PSoC. This example uses the PSoC CY8C29x66 family to implement the analog front end required for the ALS interface. The current output from the ALS is converted to a voltage signal using an external resistor. This voltage signal is then passed through a Programmable Gain Amplifier (PGA) before converting the analog signal to digital data using an ADC. An incremental ADC with up to 13-bits of resolution is used for the conversion purpose. The example project performs the following tasks:

Input

- Converts output current from the ALS to voltage signal using a resistor (R = 5.1 K, as shown in Figure 7).

Note Capacitor (C = 10 µf) is placed across the current sample resistor to reduce voltage spikes (Figure 7).

Signal Processing Stage

- Amplifies the voltage signal to the desired range using a PGA inside PSoC
- Converts the amplified signal to digital data using an incremental ADC implemented completely inside PSoC

Output

- Displays the ADC output on the LCD
- Controls the brightness of LED based on ambient light intensity (ALS output).
- Provides the converted ADC count from ALS and brightness value of the LED (PWM pulse width) to the host through I2C interface

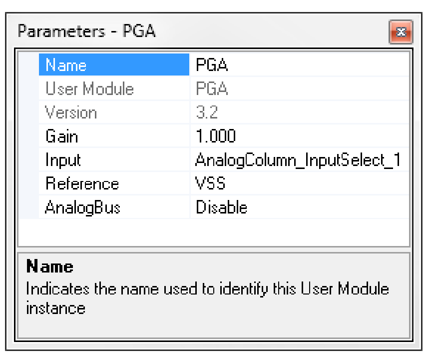
Note the project uses a periodic sleep-wake up cycle to scan the ALS every 125 ms. Refer to the project code for details.

6.1 Device Configuration

The block layout of the PSoC project is shown in Figure 6. Table 3 shows the resource requirement for this application for different PSoC families. Note that PSoC drives the V_{CC} of the ambient light sensor. This allows a zero sleep current when the sensor is completely powered down by PSoC. The project is tested using CY3210 PSoCEval1 board with CY8C29466-24PXI device. It can also be implemented in any PSoC with continuous time (CT) and switch capacitor (SC) analog blocks by configuring them to implement ADCs: CY8C21x34, CY8C21x23, CY8C24x33, CY8C24x94, CY8C27x43, or CY8C29x66.

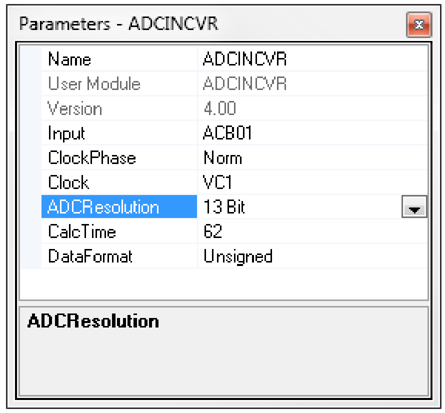
The project uses a PGA user module, which occupies one continuous time (CT) block to implement the amplifier functionality. The gain of the PGA is set to 1 to provide a full scale output of 5 V (with 5.1 KΩ resistor) when a mobile phone LED flash light is used to illuminate the ALS. The gain can be adjusted depending on the max ambient light luminosity desired from the sensor. In this case, the max luminosity is set to the mobile phone LED flash light luminosity. The reference of the PGA is set to Vss as the voltage input is referenced to Vss only. The input signal from P0[5] is routed via AnalogColumn_InputSelect_1 Mux.

Figure 3. PGA UM Configuration



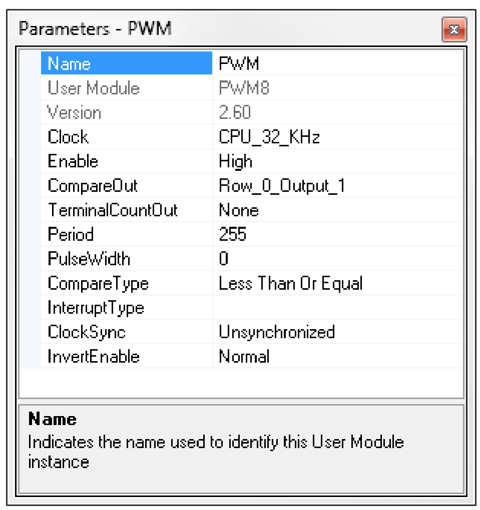
The output from the PGA feeds an incremental ADC configured to run at 13-bit resolution. The ADC uses one switched capacitor (SC) block and three basic digital blocks. Refer the ADCINCVR user module datasheet for details on the ADC parameters and their impact. With the configuration used in the project, the ADC sample rate is approximately 45.7 Samples per Second (SPS).

Figure 4. ADCINCVR UM Configuration



The project also uses a PWM8 UM clocked from the 32 KHz low frequency clock. This PWM is used to control the intensity of an LED based on the ambient light intensity. 32 KHz clock is selected as the input to make sure the PWM output is generated even during sleep. During device sleep, the high frequency clock is turned OFF to save power. The output of the PWM is routed to P1[5].

Figure 5. PWM UM Configuration



In addition to the PWM, the ambient light output is displayed in a character LCD and sent out through I2C.

Figure 6. ALS_ADC PSoC Block Layout

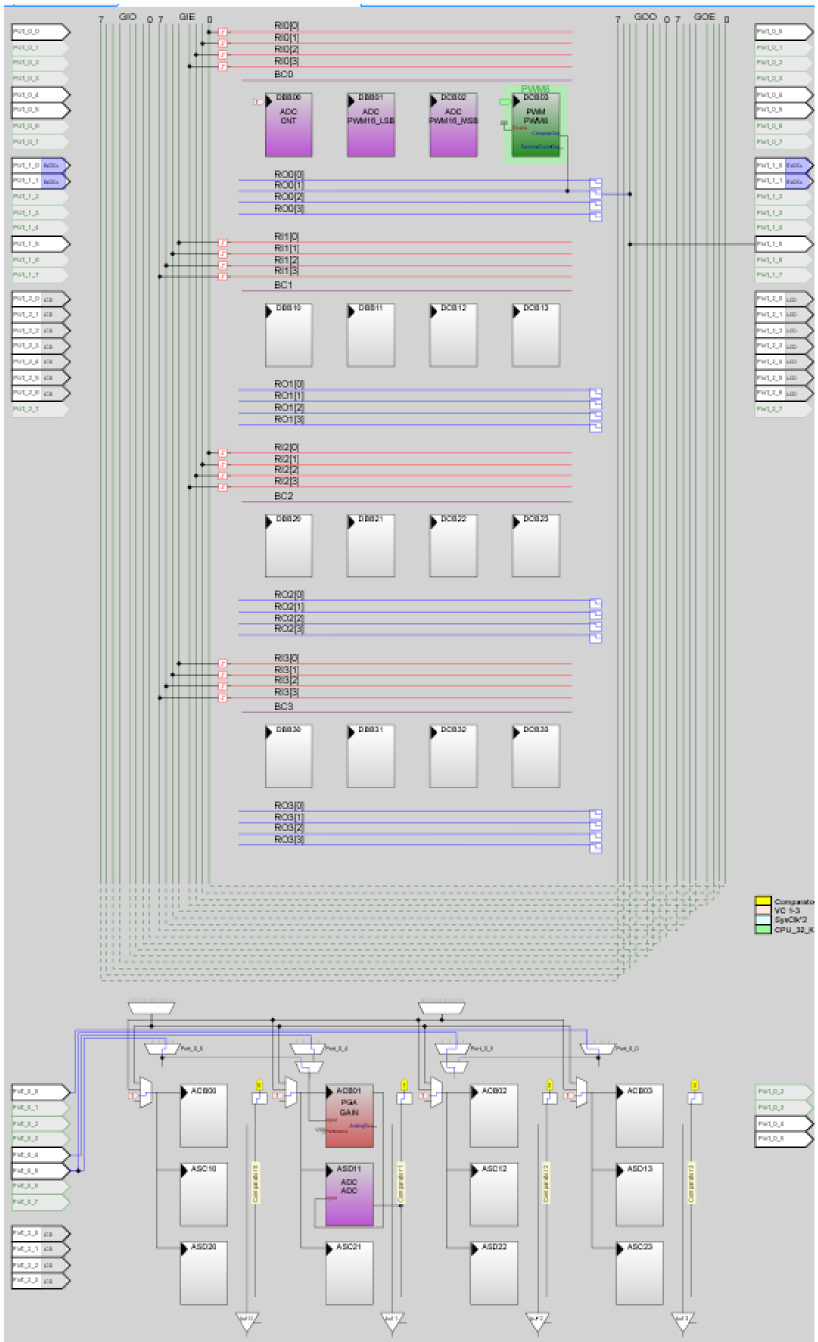


Figure 7. PSoC Design Block Diagram

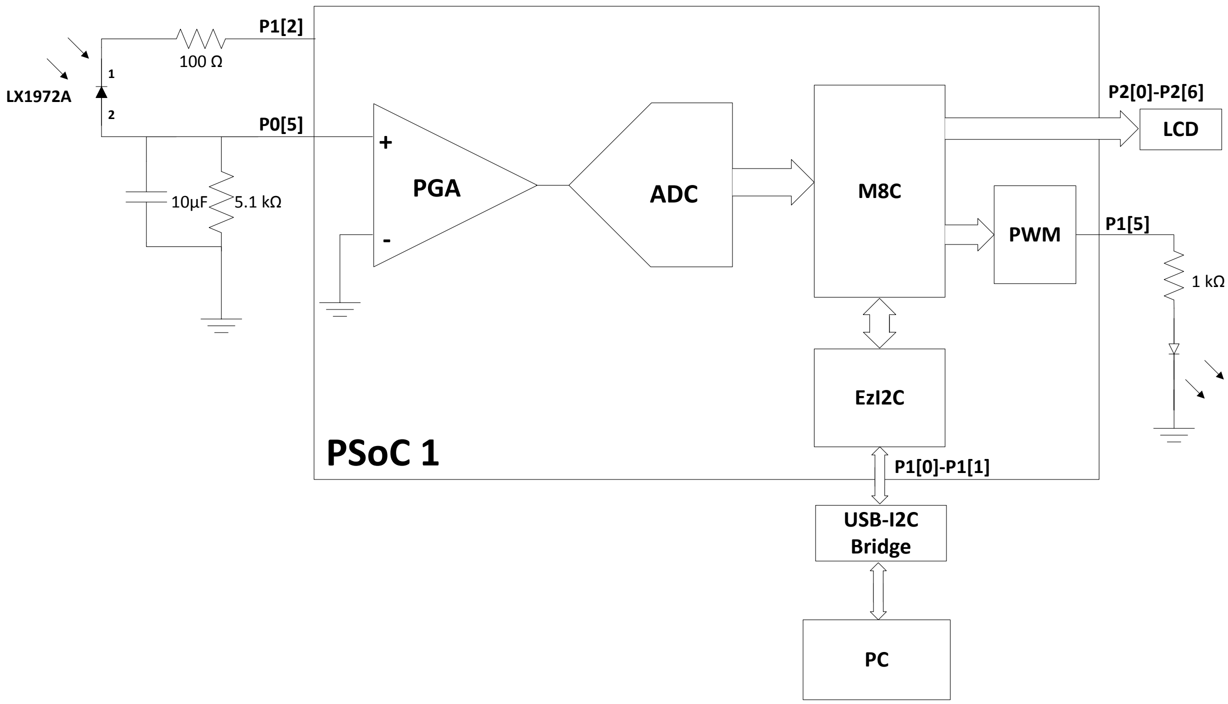


Table 3. Resource Consumption

PSoC Family	CY8C23x33	CY8C24x33	CY8C29x66	CY8C27x43	CY8C24x94	CY8C24x23A	CY8C21x34	CY8C21x23
Analog UM	PGA, ADCINC	PGA, ADCINC	PGA, ADCINC	PGA, ADCINC	PGA, ADCINC	PGA, ADCINC	ADC8	ADC8
Digital UM	PWM8	PWM8	PWM8	PWM8	PWM8	PWM8	PWM8	PWM8
DBB	4	4	4	4	4	4	4	4
DCB	-	-	-	-	-	-	-	-
ACB	1	1	1	1	1	1	1	1
ASC	1	1	1	1	1	1	1	1
ASD	-	-	-	-	-	-	-	-
Fixed resources	I2C	I2C	I2C	I2C	I2C	I2C	I2C	I2C
SW UM	Char LCD	Char LCD	Char LCD	Char LCD	Char LCD	Char LCD	Char LCD	Char LCD
I/Os used	5+6 (LCD)	5+6 (LCD)	5+6 (LCD)	5+6 (LCD)	5+6 (LCD)	5+6 (LCD)	5+6 (LCD)	5+6 (LCD)

Note

Appendix A shows the power calculations.

6.2 Hardware Requisites

1. CY3210 PSoC1Eval board with CY8C29466-24PXI device (comes with CY3210)
2. CY3217 MiniProg1 (comes with CY3210) or CY8CKIT-002 MiniProg3 for programming
3. LX1972A or equivalent Ambient light sensor
4. CY8CKIT-002 MiniProg3 for testing over I2C bridge
5. USB cables, connecting wires and 5.1 K resistor

6.3 Test Procedure

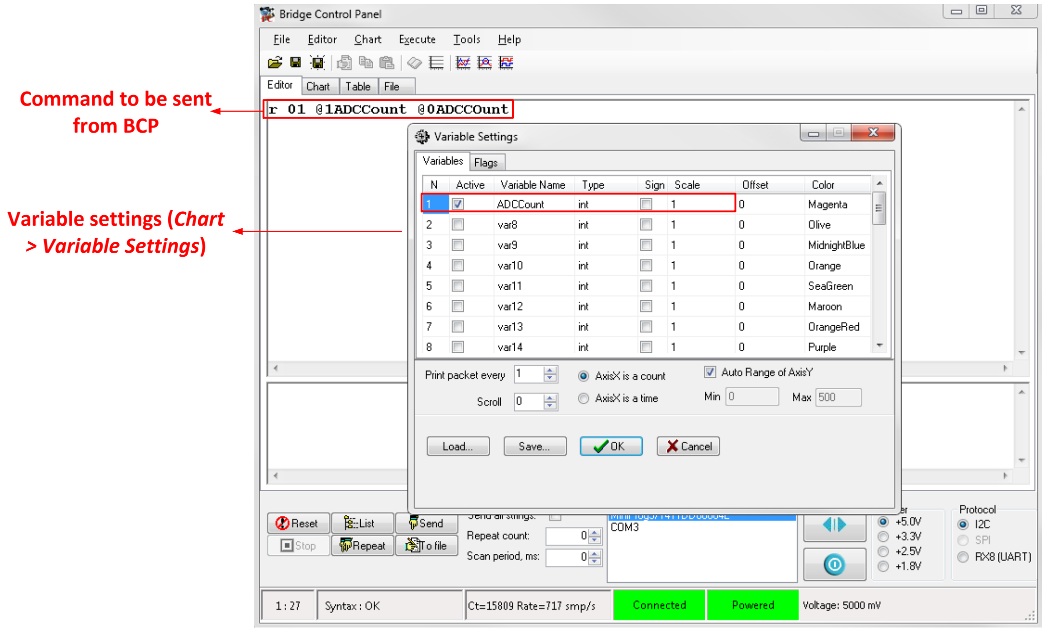
1. Connect the ALS sensor and the **5.1 KΩ** resistor as shown in Figure 7.
2. Connect the ALS sensor output to the PSoC pin **P0[5]** on the CY3210 board
3. Connect **P1[5]** to the LED. Optionally you can connect **P1[2]** to the ALS Vcc pin for turning ON/OFF the ALS through firmware.
4. Mount the **CY8C29466-24PXI** device onto CY3210
5. Connect **Miniprog1** or **Miniprog3** to the 5-Pin programming header on board
6. Open **AN52491_ALS_ADC** project and Program the device. Use Power Cycle programming option if Powering using Miniprog
7. After programming, power the board using Miniprog or power jack; the ADC count is displayed on the LCD and LED varies brightness depending on the intensity of ambient light. You can bring a mobile phone LED flash on closer to the ambient light sensor to observe the effect.
8. The project also implements the I^2C slave interface into a PSoC project through EzI2Cs user module. Any I^2C master can read this ADC count of ALS sensor from this I^2C slave device.

Table 4. Setup on the CY3210 Evaluation board

PSoC 1 Pins	CY3210 Connections	Description
P0[5]	—	Connect to ALS / 5.1 KΩ Resistor junction
P1[5]	LED1	PWM LED output
P1[2]	—	Connect to ALS VDD pin
—	Char LCD header	Character LCD output (P2[0] to P2[6]) are routed in CY3210
—	ISSP header (J11)	Connect MiniProg1 or MiniProg3 for programming
—	ISSP header (J11)	Connect MiniProg3 for I2C communication to PC; P1[0] and P1[1] are routed in CY3210

Note To see results using an I^2C to USB Bridge, comment the code that is used to put PSoC into sleep mode ("**M8C_Sleep**") Use the read command "**r 01 @1ADCCount @0ADCCount**" from the Bridge Control Panel software to read the ADC output from the ALS sensor (see Figure 8). Refer to AN2397 – CapSense Data viewing tools for details on sending and receiving commands through BCP. Make sure the I2C data rate selected (**Tools > Protocol Configuration > I2C**) does not exceed **100 KHz**.

Figure 8. Bridge Control Panel configuration



6.4 Expected Results

The LED brightness controlled by P1[5] is at its minimum when the ALS is in dark (covered with hand). The LED brightness will gradually increase when a mobile phone LED flash light is brought near it. The brightness will be at its maximum when the flash light is around 1 cm above the ALS. See Figure 10 for CY3210 board connections and output. You can refer Table 4 and Figure 7 for detailed pin connections on the CY3210.

Figure 9 shows the expected ADC output seen in Bridge control panel.

Figure 9. ADC Output in Bridge Control Panel

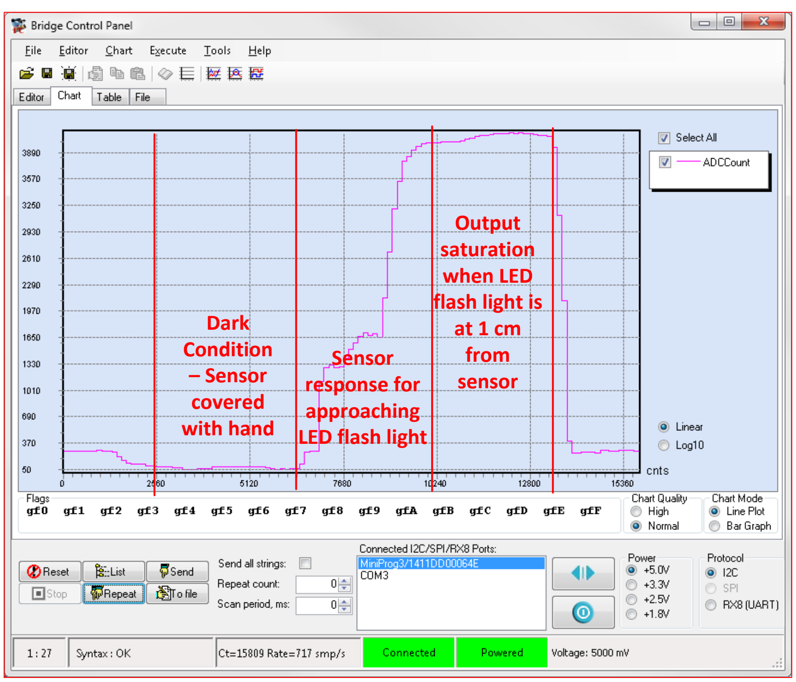
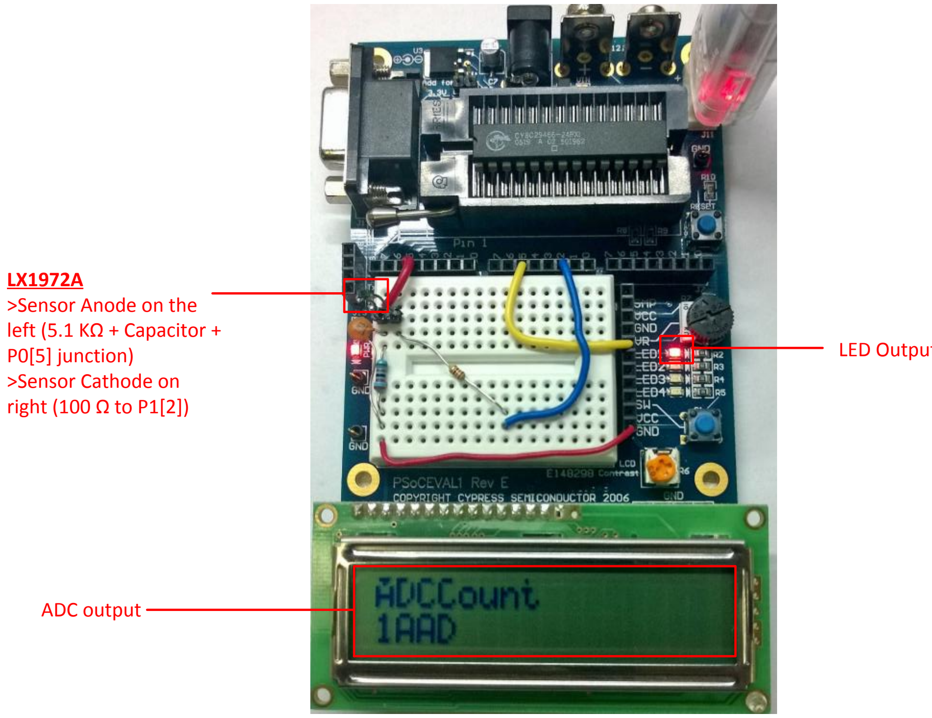


Figure 10. CY3210 Connections and Output



7 Example Project: ALS_Comparator Example

This example project demonstrates a simple application of turning on and off the lights depending on the ambient light intensity measured (such as headlamp control in cars by automatically turning on the lights in twilight or when entering a tunnel). This is done by using just a comparator. The comparator output drives a digital buffer, which in turn controls the LED ON/OFF. This example also uses the PSoC CY8C29x66 family to demonstrate the implementation. The example project performs the following tasks:

Input

- Converts output current from the ALS to voltage signal using a resistor (R = 5.1 K, as shown in Figure 7.

Note Capacitor (C = 10 µf) is placed across the current sample resistor to reduce voltage spikes (Figure 7).

Signal Processing Stage

- Compares the voltage output to a known threshold using a hardware comparator

Output

- Sends the comparator output to an LED through a digital buffer

7.1 Device Configuration

The block layout of the PSoC project is shown in Figure 12. The project is tested using CY3210 PSoC1Eval board with CY8C29466-24PXI device. It can also be implemented in any PSoC with continuous time block by configuring them to implement comparator.

The project uses comparator UM to implement the voltage comparison. The ambient light signal, P0[5], is routed to comparator non-inverting input through AnalogColumn_InputMUX_0. The threshold is kept at 210 mV (0.042 * VDD) and can be varied by changing the **RefValue** parameter of the UM. This output controls an LED connected to P1[5] through a Digital buffer UM.

Figure 11. Comparator UM configuration

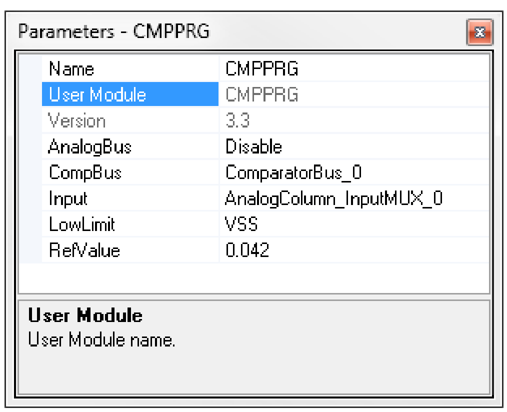
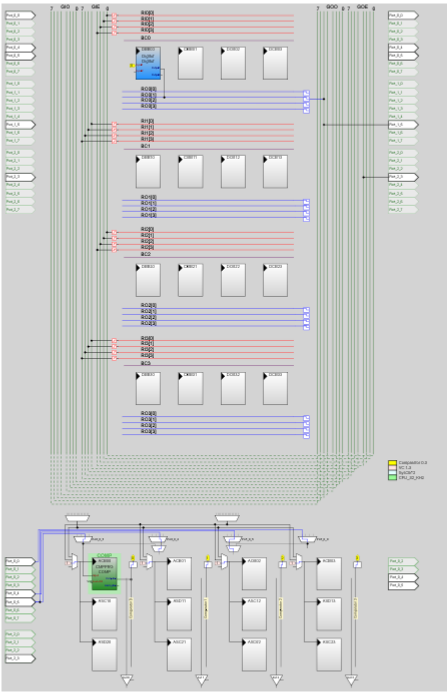


Figure 12. ALS Comparator PSoC Block Layout



7.2 Hardware Requisites

1. CY3210 PSoC1Eval board with CY8C29466-24PXI device (Comes with CY3210)
2. CY3217 MiniProg1 (comes with CY3210) or CY8CKIT-002 MiniProg3 for programming
3. LX1972A or equivalent Ambient light sensor
4. USB cables, connecting wires and 5.1 K resistor

7.3 Test Procedure

1. Connect the ALS and the **5.1 KΩ** resistor as shown in Figure 7.
2. Connect the ALS output to the PSoC pin **P0[5]** on the CY3210 board
3. Connect **P1[5]** to the LED. Optionally you can connect **P1[2]** to the ALS Vcc pin for turning ON/OFF the ALS through firmware.
4. Mount the **CY8C29466-24PXI** device onto CY3210
5. Connect **Miniprog1** or **Miniprog3** to the 5-Pin programming header on board
6. Open **AN52491_ALS_Comparator** project and Program the device. Use Power Cycle programming option
7. After programming, power the board using Miniprog or power jack; The LED connected to P1[5] should turn ON when the ALS is in dark conditions (covered with hand) and OFF when in bright conditions (mobile phone LED flash).

Table 5. Setup on the CY3210 Evaluation Board

PSoC 1 Pins	CY3210 Connections	Description
P0[5]	-	Connect to ALS / 5.1 KΩ Resistor junction
P1[5]	LED1	Comparator LED output
P1[2]	-	Connect to ALS VDD pin
-	ISSP header (J11)	Connect MiniProg1 or MiniProg3 for programming

7.4 Expected Results

LED connected to P1[5] should turn ON when the ALS is in dark conditions (covered with hand) and OFF when in bright conditions (mobile phone LED flash). The threshold at which the LED turns ON can be controlled by changing the **RefValue** parameter of the Comparator UM.

8 Summary

This application note discusses the analog signal conditioning of ambient light sensor using PSoC. The implementation proves how easy and effective it is to interface and process the sensor signals with PSoC.

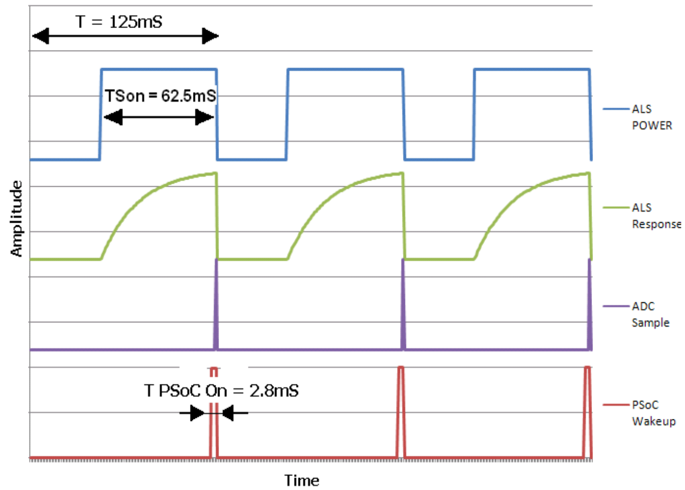
About the Author

Name:	Jaya Kathuria
Title:	Applications Engineer Sr.
Background:	Jaya Kathuria is a Senior Applications Engineer in Cypress Semiconductor's Consumer and Computation Division, focused on PSoC solutions.
Contact:	xkj@cypress.com

9 Appendix A

9.1 Power Calculations

Figure 13. Screen Shot of Output Voltage in Dark Conditions



$$I\ avg = \frac{(I\ active * Ton) + (I\ sleep * Toff)}{Ttotal}$$

- PSoC Active Current = 1.62 mA at 6 MHz
- PSoC Sleep Current = 3 µA.
- ALS Sensor Active Current = 410 µA at 100 Lx
- ALS Sensor Sleep Current = 0 A.
- See Figure 13 for time information.
- Based on the previous values and equation:
  - PSoC I average = 39.2 µA
  - Sensor I average = 205 µA.
  - I average total = 244.2

Note The sensor on time is high as its settling time is high. Settling time also varies with light intensity.

10 Document History

Document Title: AN52491 – Implementing Ambient Light Sensing Using PSoC® 1

Document Number: 001-52491

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	2678058	XKJ	03/24/2009	New application note
*A	3567112	MSUR	03/30/2012	Updated template. Completing sunset review.
*B	4732178	ASRI	05/07/2015	Updated the title to: Implementing Ambient Light Sensing Using PSoC® 1. Added a Note in Example Project: ALS_ADC Example. Replaced APDS-9002 ALS with LX1972A as APDS-9002 is obsolete and not available for purchase Updated projects to CY8C29x66 as CY8C23x33 CY3210 POD is not available for customer to test the project. Replaced SAR ADC project with incremental ADC as CY8C29x66 does not support SAR ADC and the application does not require SAR ADC Updated the example projects section to be consistent – Updated Project description, Device Configuration, Hardware Requisites, Test procedure and expected results Moved examples explanation from Appendix to main body.

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
Memory	cypress.com/go/memory
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 4 | PSoC 5LP

Cypress Developer Community

Community | Forums | Blogs | Video | Training

Technical Support

cypress.com/go/support

PSoC is a registered trademark and PSoC Creator is a trademark of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 2009-2015. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.