



**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.

## Objective

This example demonstrates the voltage comparison functionality using the LPComp Component in PSoC® 6 MCU.

## Overview

This document consists of two code examples: The first code example demonstrates how to set the Low-Power Comparator (LPComp) Component options for the internal reference voltage, set the external input from a GPIO using the LPComp driver, and read comparator result using the LPComp interrupt. The second code example demonstrates how to compare two external GPIO inputs and indicate the result using LEDs.

## Requirements

**Tool:** PSoC Creator™ 4.2

**Programming Language:** C (Arm® GCC 5.4-2016-q2-update, Arm MDK Generic)

**Associated Parts:** PSoC 6 MCU

**Related Hardware:** CY8CKIT-062-BLE Pioneer Kit

## Design

### Code Example 1

This code example features one LPComp, one Global Signal Reference (GSR) with an interrupt, one status LED, one GPIO for an external input, and one potentiometer on the  $V_{plus}$  pin, as Figure 1 shows.

Figure 1. Comparing an External Voltage and the Vref Using LPComp

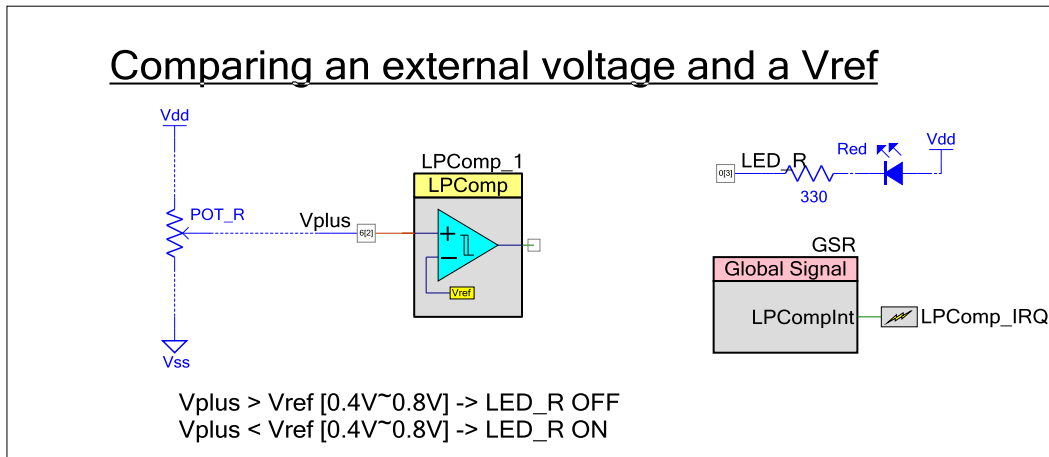
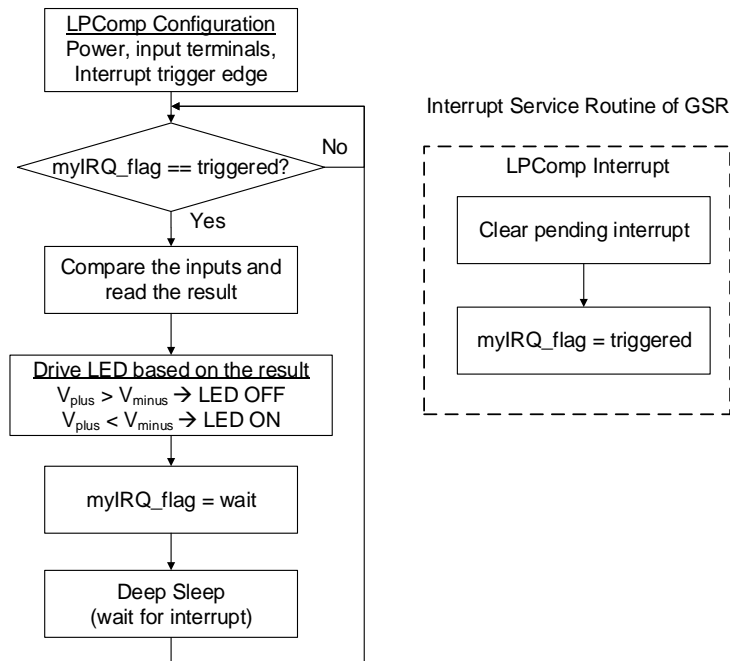


Figure 2 shows the flow of the firmware. `main()` compares the LPComp Component inputs when the interrupt flag changes to the triggered state, and then sets the flag to the wait state and goes into Deep Sleep to wait for the next interrupt.

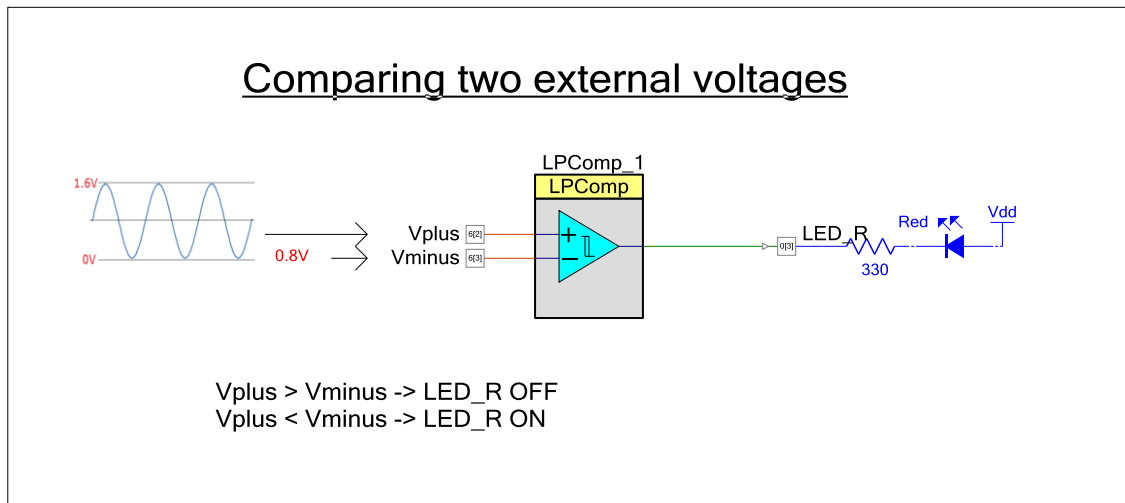
Figure 2. Interrupt-Based Voltage Comparison Flowchart



### Code Example 2

This code example features the following Components: one LPComp, one GPIO for the status LED, and two GPIOs for analog inputs. The positive terminal ( $V_{plus}$ ) connects to the output of a function generator to input the sine wave (in this example, the sine wave has a 0.8-V offset, 1.6-Vpp and 1-Hz frequency). The negative terminal ( $V_{minus}$ ) connects the reference voltage (in this example, the voltage is 0.8 V.), as Figure 3 shows.

Figure 3. Comparing Two External Voltages Using LPComp



The comparison result from the comparator out terminal is routed directly to the LED\_R pin. Because of this routing, LED\_R indicates the comparison result without any additional firmware code. Therefore, main() of this example is empty.

## Design Considerations

The status LED (LED\_R) shows the voltage comparison result. If the positive terminal input is higher than the negative terminal input, the red LED is OFF; it is ON otherwise.

These examples place the GPIOs, LPComp, and Global Signal Reference as shown in Table 1. By placing and compiling the Component, PSoC Creator copies the necessary Peripheral Driver Library (PDL) into project folders.

## Components

Table 1 lists the PSoC Creator Components used in this example, as well as the placement used by each.

Table 1. List of PSoC Creator Components

Component	Instance Name	Hardware Resource
Low Power Comparator	LPComp_1	1 Component
Analog Pin	Vplus	1 GPIO
Analog Pin	Vminus	1 GPIO (Code example 2 only)
Digital Output Pin	LED_R	P0_3
Global Signal Reference	GSR	1 Component (Code example 1 only)
Interrupt	LPComp_IRQ	1 Component (Code example 1 only)

## LPComp

Figure 4. LPComp Component Settings for Example 1

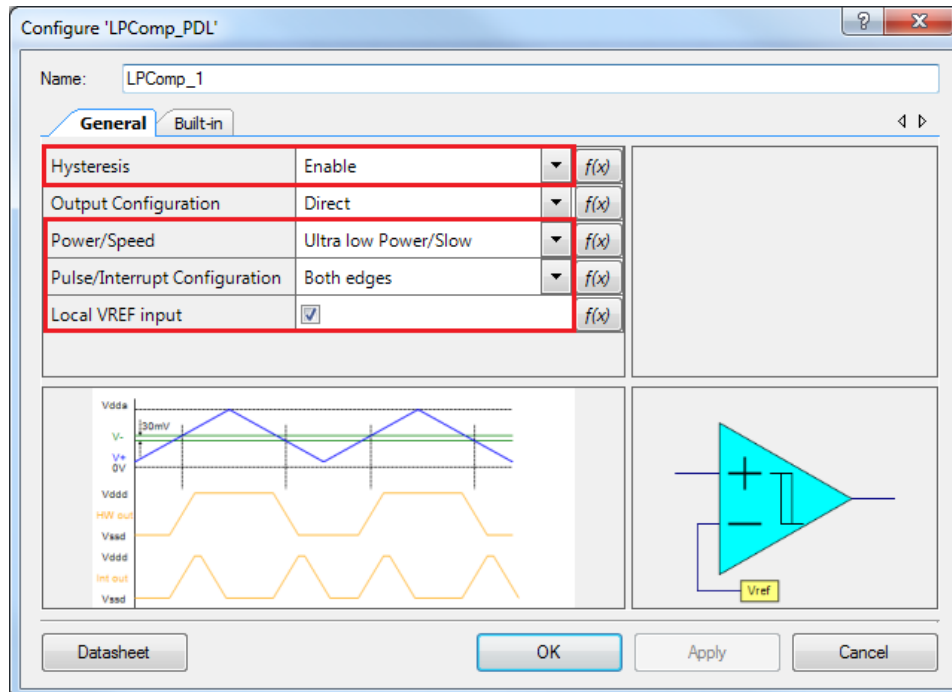
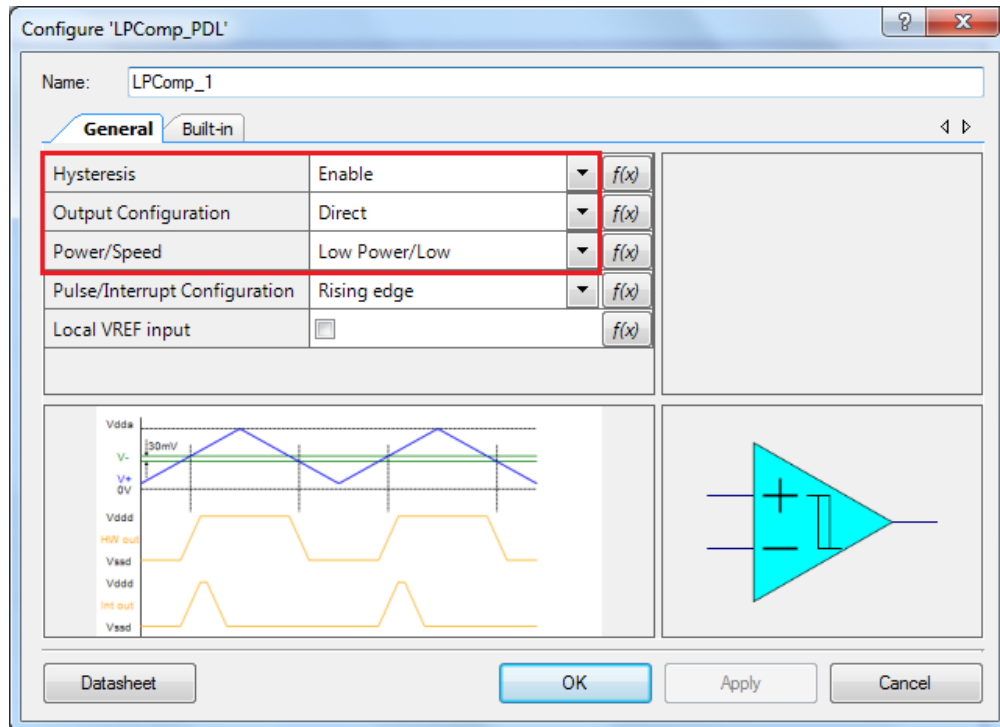


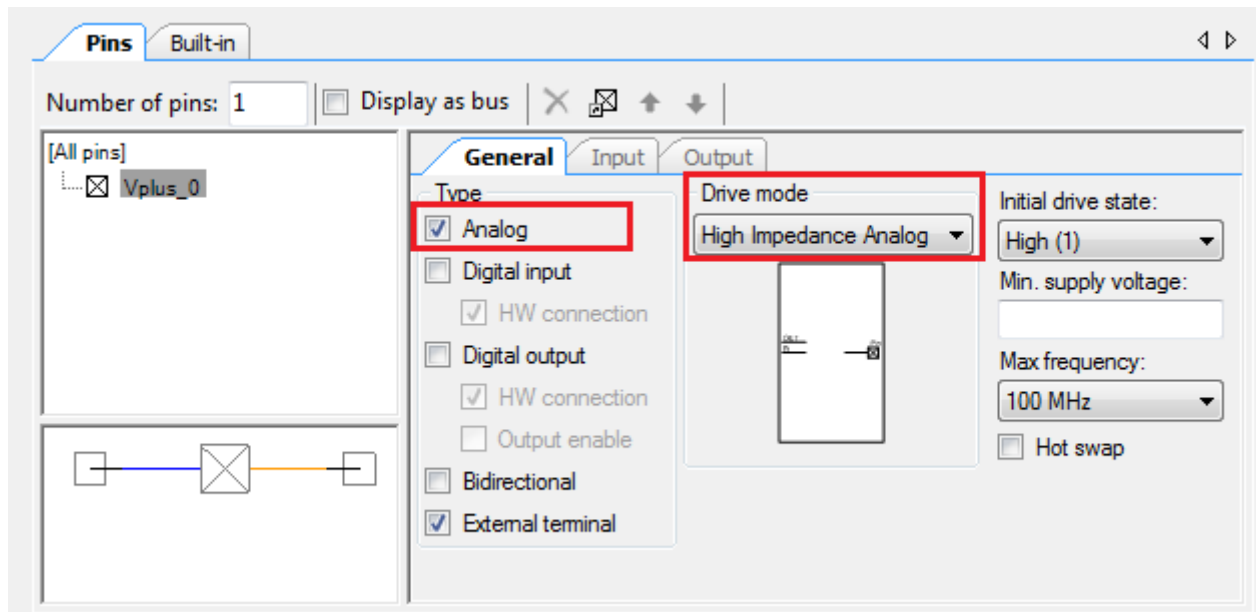
Figure 5. LPComp Component Settings for Example 2



**Analog Pin**

Figure 6 shows the setting for the analog input pin.

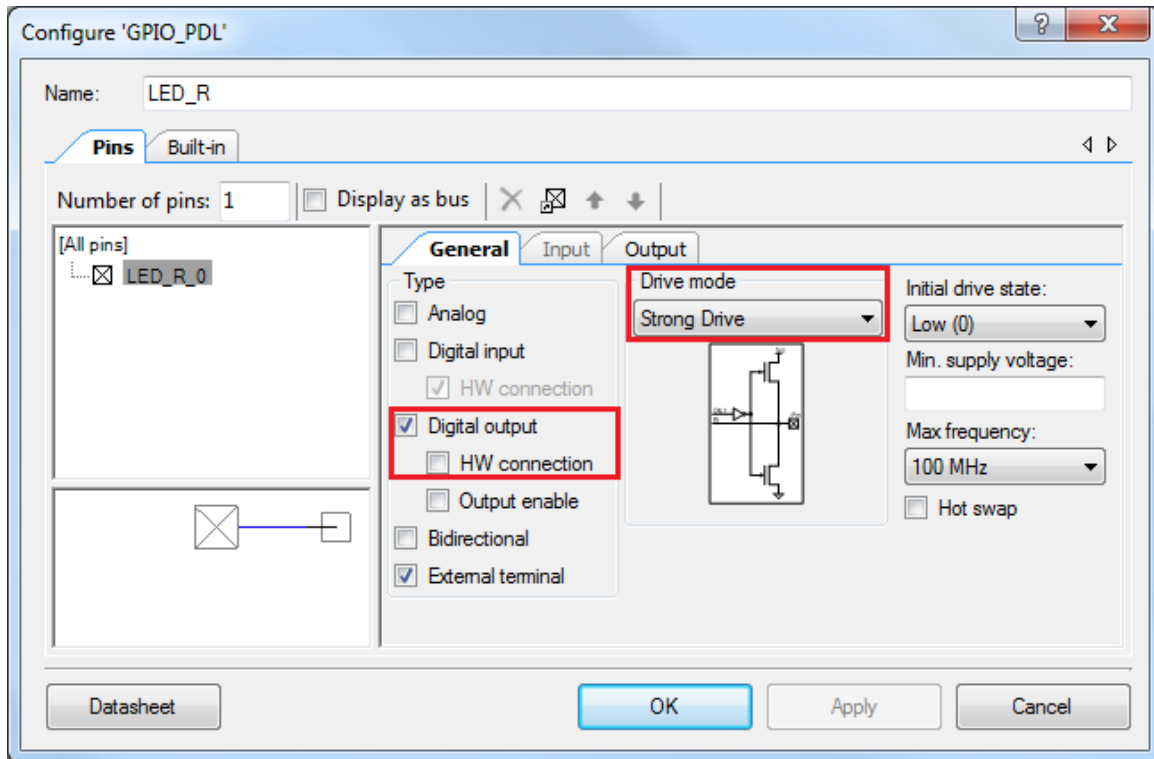
Figure 6. Analog Pin Configuration



**Status LED pin**

Figure 7 shows the settings for the status LED control pin. Set the drive mode as “Strong Drive” and the pin type as “Digital output”, and uncheck HW connection.

Figure 7. Status LED Pin Configuration

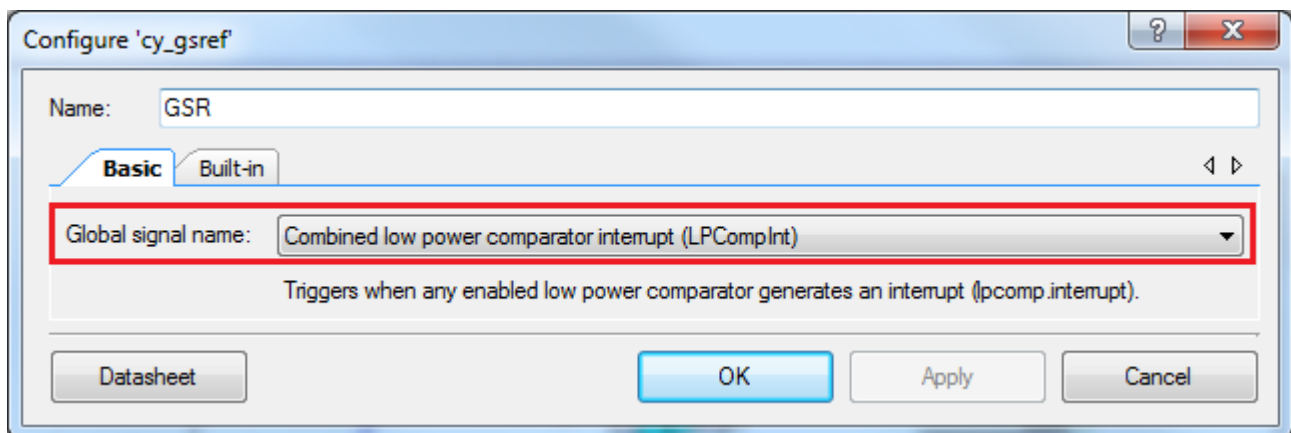


### Global Signal Reference

Set the Global signal name to “Combined low power comparator interrupt (LPCompInt)”, as Figure 8 shows.

This is applicable only for code example 1.

Figure 8. Global Signal Reference Configuration



## Design-Wide / Global Resources

Table 2 shows the pin assignments for the switch and LEDs of CY8CKIT-062-BLE.

Table 2. DWR Pin Assignment Table

Component	Instance Name	Pin
Digital Output Pin	LED_R	P0[3]
Analog Pin	Vplus	P6[2]
Analog Pin	Vminus	P6[3] (Code example 2 only)

## Operation

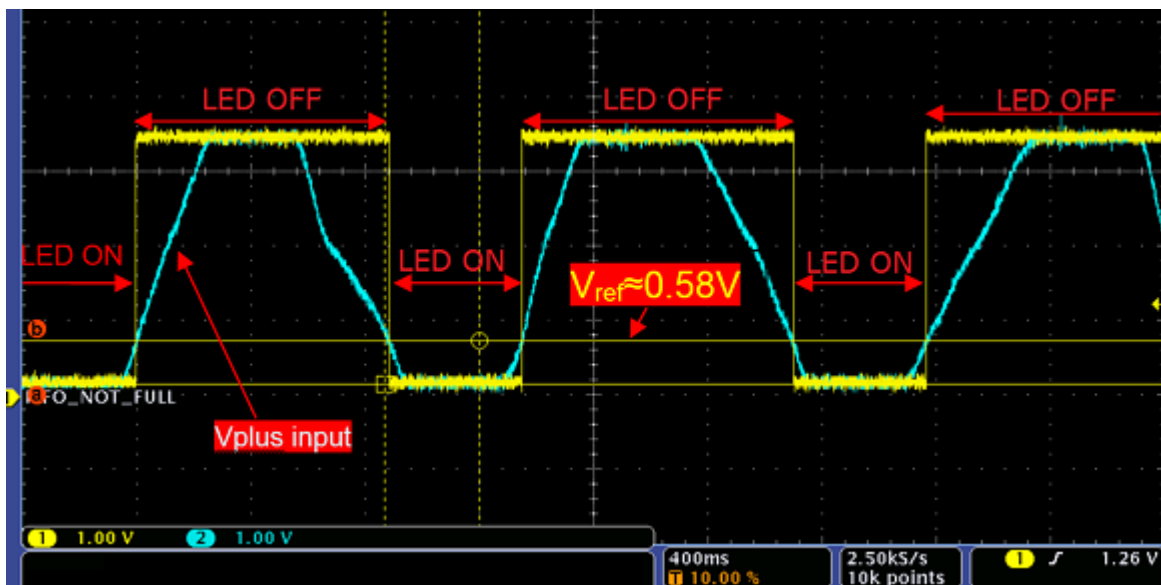
Follow the instructions that came with your kit to make sure that your kit is connected to your PC.

### Code Example 1

1. Build the “CE218472\_LPComp\_Ext\_Vref” project and program the CY8CKIT-062-BLE kit.
2. Place a potentiometer on P6[2] to change the  $V_{plus}$  input voltage.
3. Turn the knob of the potentiometer until the  $V_{plus}$  input is high ( $=V_{dd}$ ).
4. Confirm that LED\_R is OFF.
5. Turn the knob of the potentiometer until the  $V_{plus}$  input is low ( $=GND$ ).
6. Confirm that LED\_R is ON.

Figure 9 shows the plots of LED\_R and Vplus. It shows that LED\_R turns ON when Vplus goes lower than 0.58 V; otherwise it turns OFF.

Figure 9. Plots of LED\_R and the External Input (Vplus)



### Code Example 2

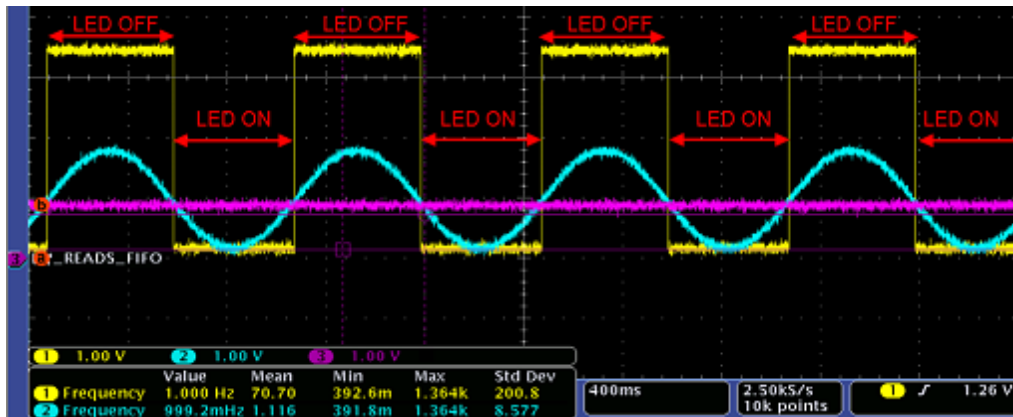
1. Build the “CE218472\_LPComp\_Two\_Ext” project and program the CY8CKIT-062-BLE kit.
2. Connect a function generator output to P6[2].
3. Configure the function generator output for generating a sine wave.

The sine wave should be equal or higher than 0 V. (In this code example, it uses 0 V – 1.6 V and 1 Hz)

4. Connect a reference voltage ( $V_{\text{minus}}$ ) to P6[3].  
This voltage should be higher than the minimum voltage of the sine wave and lower than the maximum voltage of the sine wave. (In this code example, it uses 0.8 V)
5. Enable the output of the function generator.
6. Confirm that LED\_R toggles following the sine wave frequency.

Figure 10 shows plots of LED\_R, Vplus and  $V_{\text{minus}}$ . It shows that LED\_R turns ON when Vplus is lower than  $V_{\text{minus}}$ ; otherwise it turns OFF.

Figure 10. Plots of LED\_R and Two External Inputs (Vplus and  $V_{\text{minus}}$ )



## Related Documents

Application Notes	
<a href="#">AN210781 – Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity</a>	Introduction of PSoC 6 MCU with Bluetooth Low Energy (BLE)
PSoC Creator Component Datasheets	
<a href="#">Pins</a>	Supports connection of hardware resources to physical pins
<a href="#">Low Power Comparator</a>	Supports low power comparators
<a href="#">SysInt</a>	Provides SysInt component settings
Device Documentation	
<a href="#">PSoC 6 MCU: PSoC 63 with BLE Datasheet</a>	<a href="#">PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual</a> <a href="#">PSoC 6 MCU: PSoC 63 with BLE Registers Technical Reference Manual</a>
Development Kit (DVK) Documentation	
<a href="#">CY8CKIT-062-BLE Pioneer Kit</a>	



## Document History

Document Title: CE218472 – PSoC 6 MCU: Comparing External Voltages Using a Low-Power Comparator

Document Number: 002-18472

Revision	ECN	Orig. of Change	Submission Date	Description of Change
*A	5975021	AJYA	11/24/2017	Initial Public Release

## 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

Arm® Cortex® Microcontrollers	<a href="http://cypress.com/arm">cypress.com/arm</a>
Automotive	<a href="http://cypress.com/automotive">cypress.com/automotive</a>
Clocks & Buffers	<a href="http://cypress.com/clocks">cypress.com/clocks</a>
Interface	<a href="http://cypress.com/interface">cypress.com/interface</a>
Lighting & Power Control	<a href="http://cypress.com/powerpsoc">cypress.com/powerpsoc</a>
Memory	<a href="http://cypress.com/memory">cypress.com/memory</a>
PSoC	<a href="http://cypress.com/psoc">cypress.com/psoc</a>
Touch Sensing	<a href="http://cypress.com/touch">cypress.com/touch</a>
USB Controllers	<a href="http://cypress.com/usb">cypress.com/usb</a>
Wireless/RF	<a href="http://cypress.com/wireless">cypress.com/wireless</a>

### PSoC® Solutions

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6](#)

### Cypress Developer Community

[Forums](#) | [Projects](#) | [Videos](#) | [Blogs](#) | [Training](#) | [Components](#)

### Technical Support

[cypress.com/support](http://cypress.com/support)

PSoC is a registered trademark and PSoC Creator is a trademark of Cypress Semiconductor Corporation. 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

© Cypress Semiconductor Corporation, 2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit [cypress.com](http://cypress.com). Other names and brands may be claimed as property of their respective owners.