



**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 code example demonstrates how to implement low-power CapSense® buttons with an average current consumption of 5  $\mu$ A per button.

## Overview

This code example implements two CapSense buttons using [CY8CKIT-041-40XX PSoC 4 S-Series Pioneer Kit](#), as shown in [Figure 1](#). The left button is used to control the onboard RGB LED color and the right button is used to control the brightness of the RGB LED. Using the low-power modes available in the PSoC® 4000S device, an average current of 5  $\mu$ A per button is achieved when the touch is not detected.

Figure 1. Left and Right Buttons on the Kit



## Requirements

**Tool:** PSoC Creator™ 4.0 and later versions

**Programming Language:** C (ARM® GCC 4.9.3)

**Associated Parts:** All PSoC 4000S parts

**Related Hardware:** [CY8CKIT-041-40XX PSoC 4 S-Series Pioneer Kit](#)

## Design

[Figure 2](#) and [Figure 3](#) show the PSoC Creator schematics of this code example. This code example uses the CapSense, PWM, Pin, Clock, and EZI2C Slave Components.

The CapSense Component is configured to scan two self-capacitance-based button widgets – left button and right button and a ganged widget. The two button sensors are combined and scanned as a ganged sensor. The EZI2C Slave Component is used to monitor the sensor data on the PC using the CapSense Tuner available in the PSoC Creator integrated design environment (IDE).

The PWM Component controls the brightness of the RGB LED by driving a pseudo-random PWM signal. A pseudo-random PWM signal spreads the energy of the PWM signal at different frequencies so that it is easy to filter the higher-order harmonics, if required.

Figure 4 shows the flowchart for the code example. To reduce the power consumed by the PSoC device and provide an optimum touch response, this code example implements two modes: Fast Scan and Slow Scan. When the user is interacting with the buttons, the PSoC device is in the Fast Scan mode and when the user is not interacting with the buttons for a specific duration, the Slow Scan mode is used.

In the Fast Scan mode, both button sensors are scanned at a refresh rate of 50 Hz (or a scan interval of 20 ms), and the RGB LED is driven based on the button status. The PSoC device is put into the CPU Sleep mode after the CapSense data is processed. The watchdog timer is used to periodically wake up the device from the Sleep mode. This mode provides an optimum touch response, but consumes a higher power when compared to the Slow Scan mode.

In the Slow Scan mode, both the button sensors are ganged and scanned at a refresh rate of 10 Hz (or a scan interval of 100 ms). The RGB LED is turned OFF, and the PSoC device is put into the Deep Sleep mode periodically. The Slow Scan mode consumes a lower average power of 5 µA per button, but with a slower touch response. Once touch is detected in the Slow Scan mode, the PSoC device switches to the Fast Scan mode to provide the optimum touch response at the expense of a higher power consumption.

In the Fast Scan mode, when the left button is touched, the color of the RGB LED is changed in the following order: Red→ Green→ Blue→ Red. When right button is touched, the brightness of the RGB LED is varied in the order: Low→ Medium→ High→Low.

Figure 2. TopDesign – CapSense Page

### CE210488 Low Power CapSense Buttons

This code example demonstrates how to implement a low-power CapSense button with an average current consumption of 5µA per button.

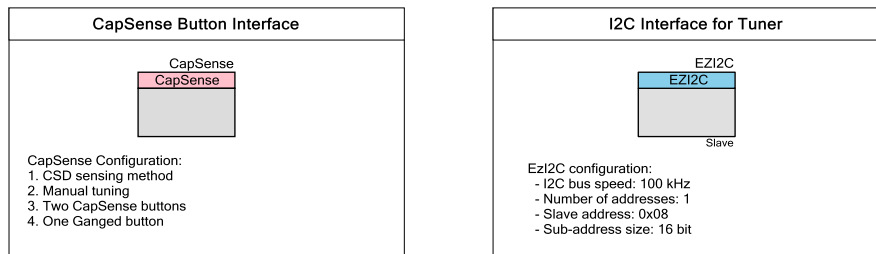


Figure 3. TopDesign – RGB LED Drive Page

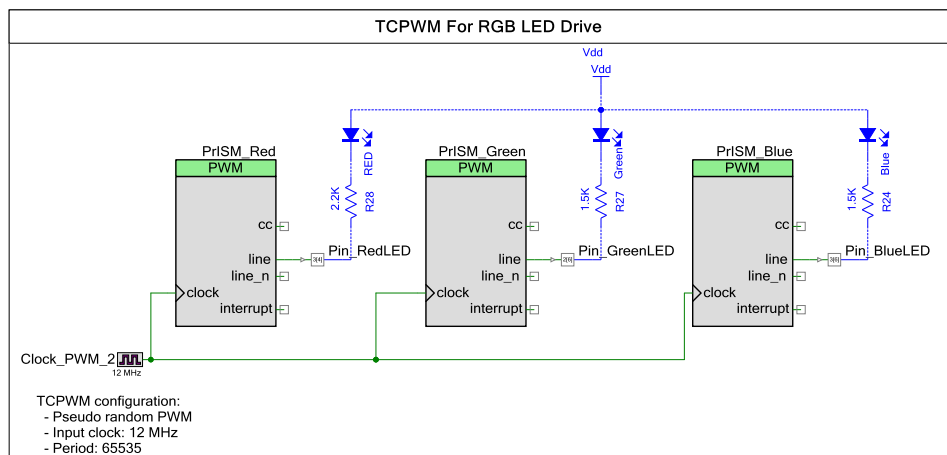
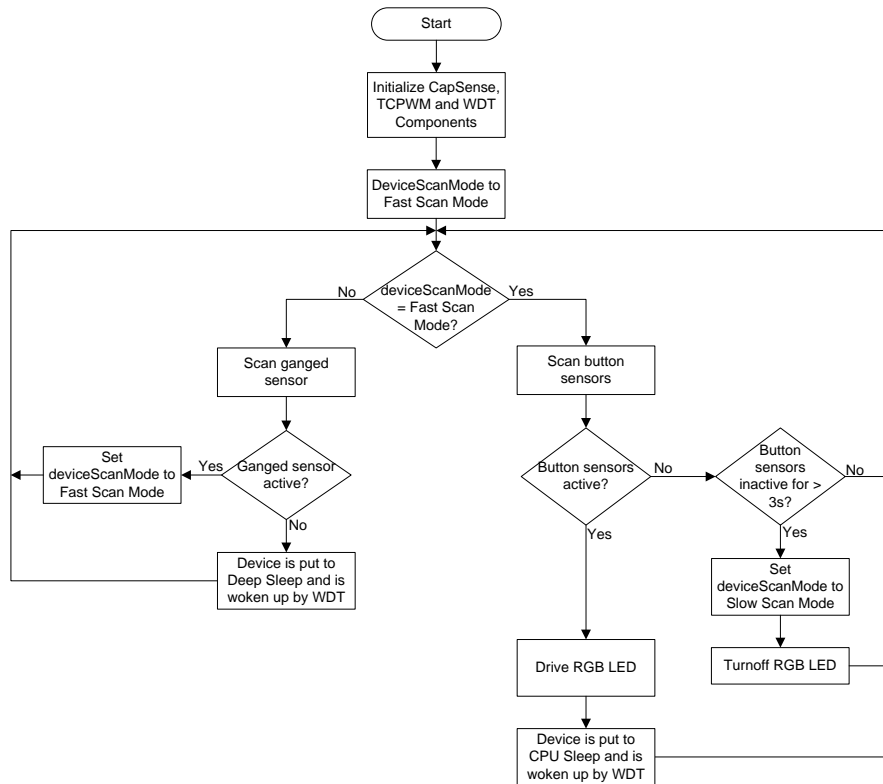


Figure 4. Flow Chart



## Design Considerations

This code example is designed to run on the [CY8CKIT-041-40XX PSoC 4 S-Series Pioneer Kit](#) with the PSoC 4000S device. To port the design to other PSoC 4 devices and kits, you must change the target device in Device Selector, change the pin assignments in the .cydwr settings, and re-tune the CapSense sensors. For the tuning procedure, see [AN85951 – PSoC 4 and PSoC Analog Coprocessor CapSense Design Guide](#).

The response time for the first touch after the sensor is inactive is around 100 ms because the refresh rate is set to 100 ms in the Slow Scan mode to achieve an average current of 5 µA per button. You can configure the refresh rate by changing the macro LOOP\_TIME\_SLOWSCANMODE in the main.c file.

## Hardware Setup

The code example works with the default settings on the [CY8CKIT-041-40XX PSoC 4 S-Series Pioneer Kit](#). If the settings are different from the default values, see the "Switches Default Position" table in the kit guide to reset to the default settings.

## Software Setup

The code example does not require any special software considerations.

## PSoC Creator Components

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

Table 1. List of PSoC Creator Components

Component	Instance Name	Version	Hardware Resources
CapSense	CapSense	v3.10	CSD, 3 GPIO pins
EZ12C Slave (SCB mode)	EZ12C	v3.20	SCB, 2 GPIO pins
Clock	Clock_PMW_2	v2.20	1 Clock Divider
PWM (TCPWM mode)	PrISM_Red, PrISM_Green, PrISM_Blue	v2.10	1 TCPWM each
Digital Output Pin	Pin_BlueLED , Pin_GreenLED, Pin_RedLED	v2.20	1 GPIO pin each

## Parameter Settings

### CapSense

Figure 5, Figure 6, and Figure 7 show the CapSense Component settings that are changed from the default values. See the [CapSense Component datasheet](#) for additional information.

Figure 5: CapSense Component – Basic Tab Configuration

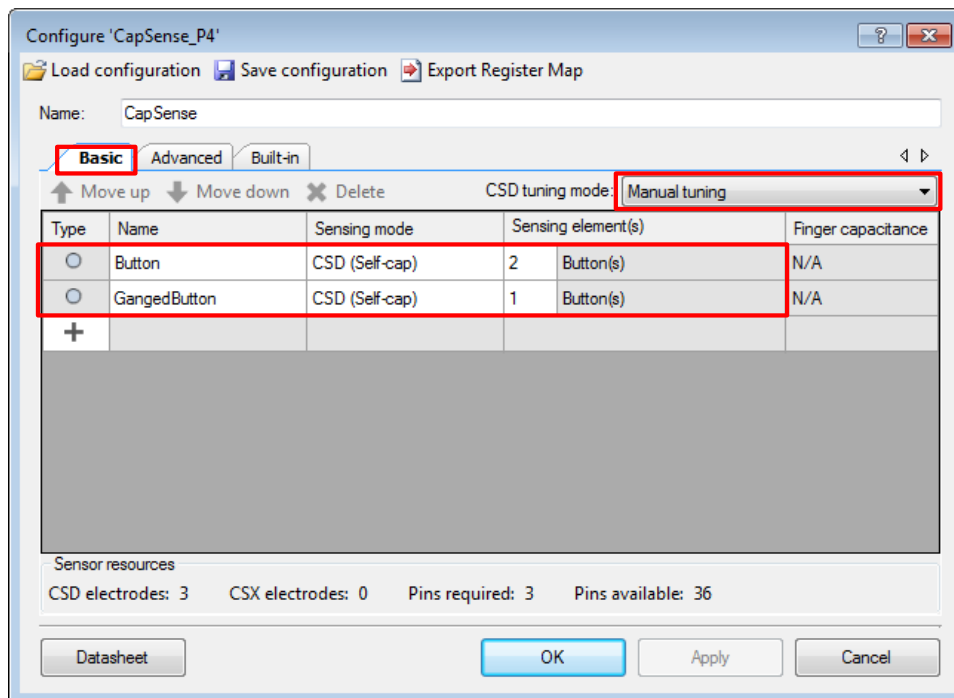


Figure 6. CapSense Component – Advanced Tab CSD Settings

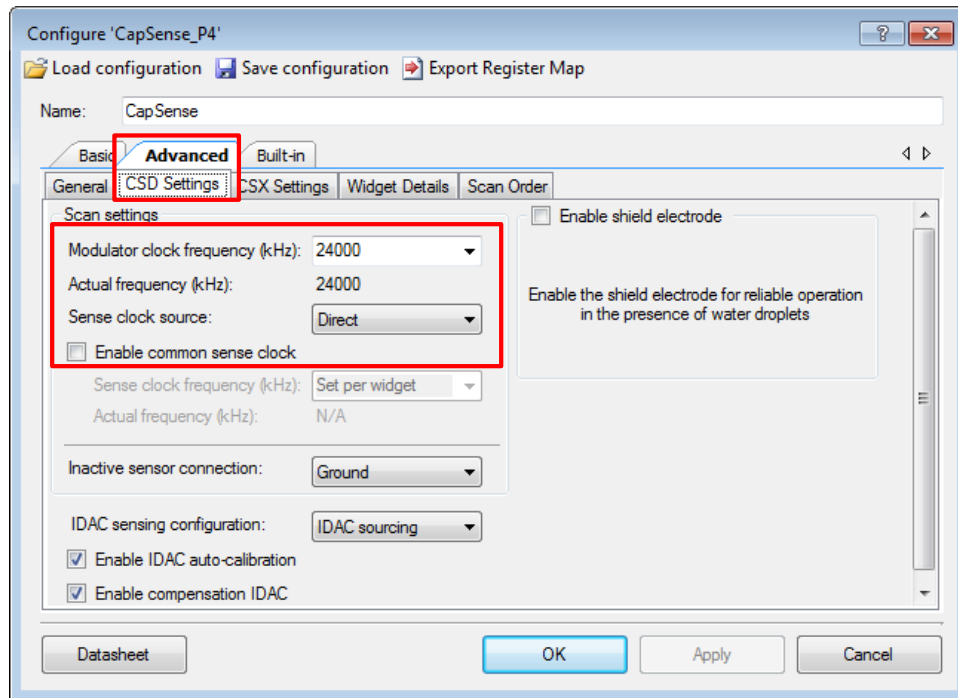
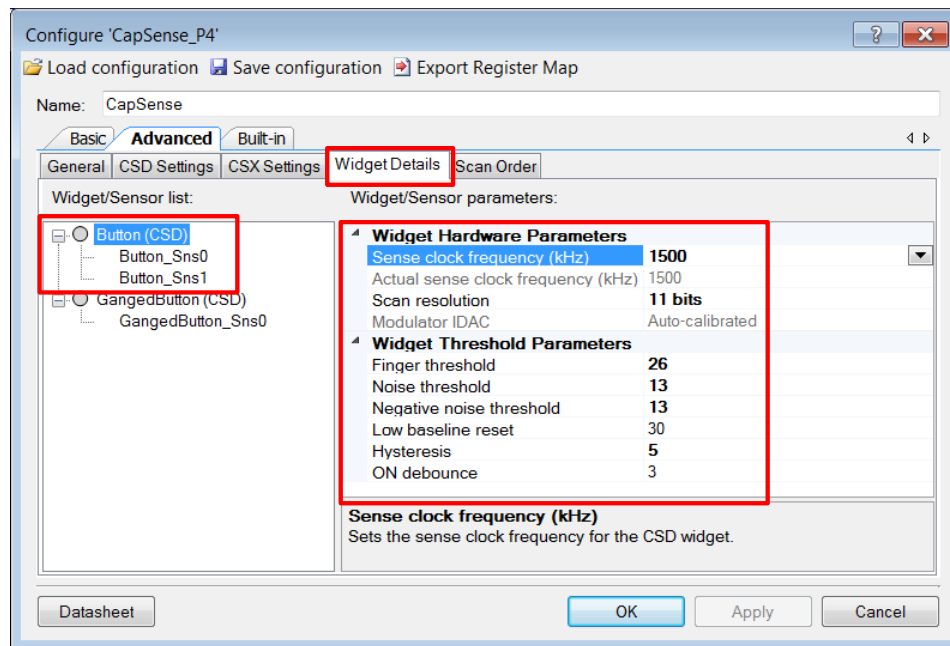


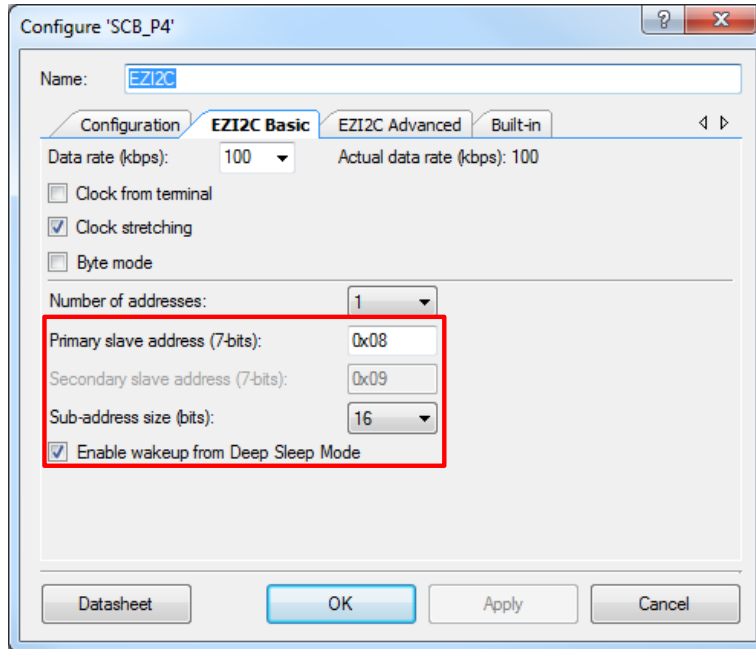
Figure 7. CapSense Component – Advanced Tab Widget Details



### EZI2C Slave

Figure 8 shows the non-default EZI2C Slave Component settings. See the [SCB Component datasheet](#) for additional information.

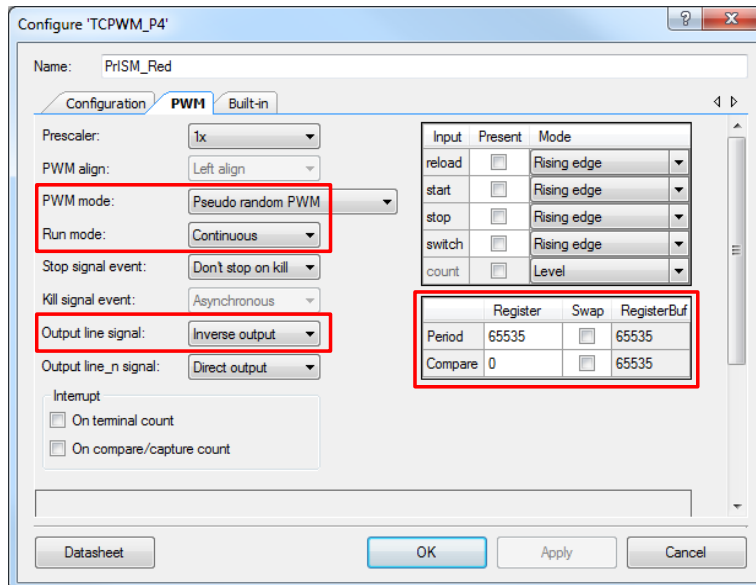
Figure 8. EZI2C Slave Component Basic Settings



### PWM

Figure 9 shows the non-default PWM Component settings. See the [TCPWM Component datasheet](#) for additional information.

Figure 9. PWM Component Configuration



## Design-Wide

Figure 10 and Figure 11 show the non-default .cydwr settings for the project.

Figure 10. .cydwr Pins Tab Settings

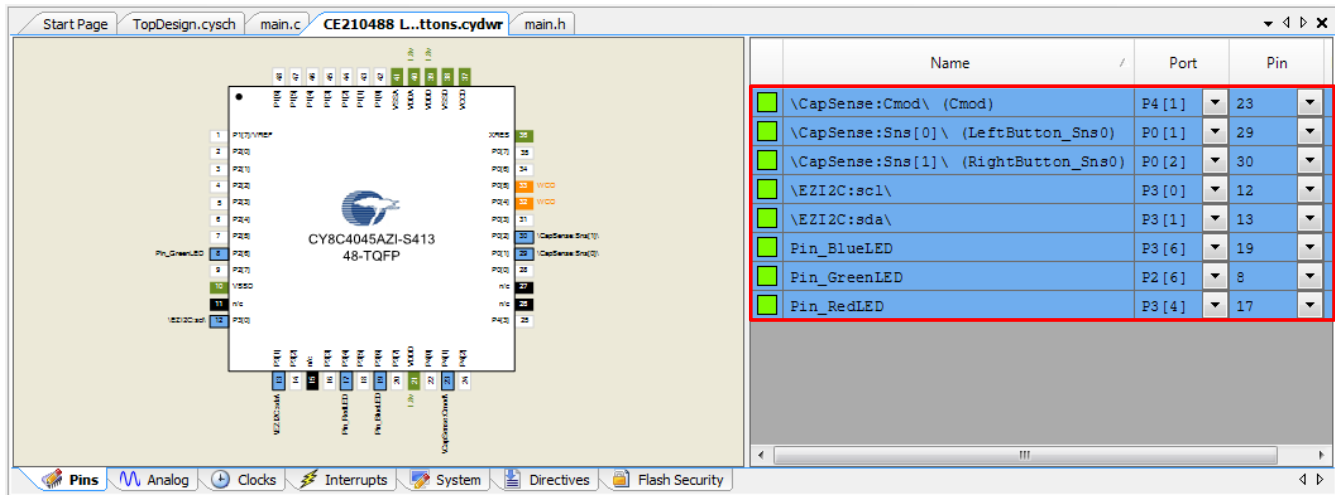
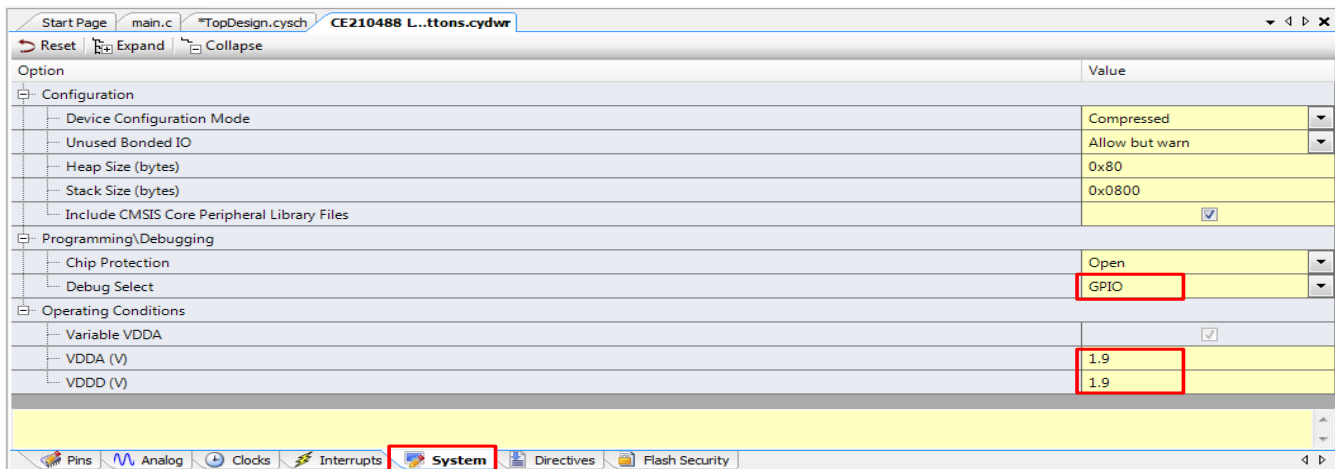


Figure 11. .cydwr System Tab Settings



**Note:** For PSoC 4 S-Series device, the CapSense  $V_{REF}$  voltage is set based on the VDDA setting in the cydwr tab as per the following table.

Table 2. CapSense  $V_{REF}$  Values Based on VDDA Setting

VDDA (V)	VREF (V)
< 2.7	1.2
2.7 to 4.8	2.1
>= 4.8	4.2

If VDDA is set to 1.9 V in the .cydwr tab,  $V_{REF}$  is set to 1.2 V. This  $V_{REF}$  voltage ensures that the CapSense tuning parameters do not vary with respect to VDDA thereby avoiding retuning of sensors.



## Operation

Follow the steps below to test the project:

1. Select the *CE210488 LP CapSense Buttons.cywrk* file in the PSoC Creator Start Page, under **Examples and Kits > Kits > CY8CKIT-041-40XX**. Select a location to save the code example.
2. Build the project (**Build > Build CE210488 LP CapSense Buttons**).
3. Connect the PSoC 4 S-Series Pioneer Kit to your computer using the USB cable provided.
4. Program the PSoC 4000S device (**Debug > Program**). See the kit guide for details on programming the kit.
5. Touch the left button and observe that the Red LED is turned ON. Upon repeated touches, the RGB LED turns on in the following sequence: Red→Green→Blue→Red.
6. Touch the right button and observe that the RGB LED color brightness changes. Upon repeated touches, the brightness switches among three levels: Min→Mid→Max.
7. Connect an ammeter between P4.VDD and VDD test points on the main board to measure the PSoC 4000S device current consumption. See the “Current Measurement Switch” section in the kit guide for complete details on power measurement steps.
8. Release the finger from buttons and wait for three seconds. Notice that the average current is about 5  $\mu$ A per button.  
**Note:** At 5 V, the average current consumption is much higher than 5  $\mu$ A. This is because the VDDA value in the *.cydwr* settings is set to 1.9 V instead of the actual operating voltage. See the “Low Voltage Analog Boost Clocks” section in the [PSoC 4 System Reference Guide](#) for more information.
9. Touch the buttons and notice that the PSoC 4000S current consumption increases to 3 mA because the LEDs are turned ON.

## Upgrade Information

The code example is updated to the latest version of PSoC Creator and therefore does not require an upgrade.

## Related Documents

Table 3 lists the relevant application notes, code examples, PSoC Creator Component datasheets, device documentation, and development kit (DVK) documentation.

Table 3. Related Documents

Application Notes		
<a href="#">AN79953</a>	Getting Started with PSoC 4	Describes PSoC 4, and how to build your first PSoC Creator project.
<a href="#">AN85951</a>	PSoC 4 and PSoC Analog Coprocessor CapSense Design Guide	Describes PSoC 4 and PSoC Analog Coprocessor CapSense Component tuning
Code Examples		
<a href="#">CE210291</a>	PSoC 4 CapSense One Button	
<a href="#">CE210290</a>	PSoC 4 CapSense Low-Power Ganged Sensor	
PSoC Creator Component Datasheets		
<a href="#">CapSense</a>	Supports capacitive touch sensing	
<a href="#">PWM</a>	Supports 16-bit fixed-function Pseudo random PWM implementation	
<a href="#">EZI2C Slave</a>	Supports I2C slave operation	
<a href="#">Pins</a>	Supports connection of hardware resources to physical pins	
<a href="#">Clock</a>	Supports local clock generation	
Device Documentation		
<a href="#">PSoC 4000S Family Datasheet</a>		
<a href="#">PSoC 4000S Family PSoC 4 Architecture Technical Reference Manuals</a>		
Development Kit (DVK) Documentation		
<a href="#">CY8CKIT-041-40XX PSoC 4 S-Series Pioneer Kit</a>		

## PSoC Resources

Cypress provides a wealth of data at [www.cypress.com](http://www.cypress.com) to help you to select the right PSoC device for your design, and quickly and effectively integrate the device into your design. For a comprehensive list of resources, see [KBA86521](#), [How to Design with PSoC 3](#), [PSoC 4](#), and [PSoC 5LP](#). The following is an abbreviated list for PSoC 4:

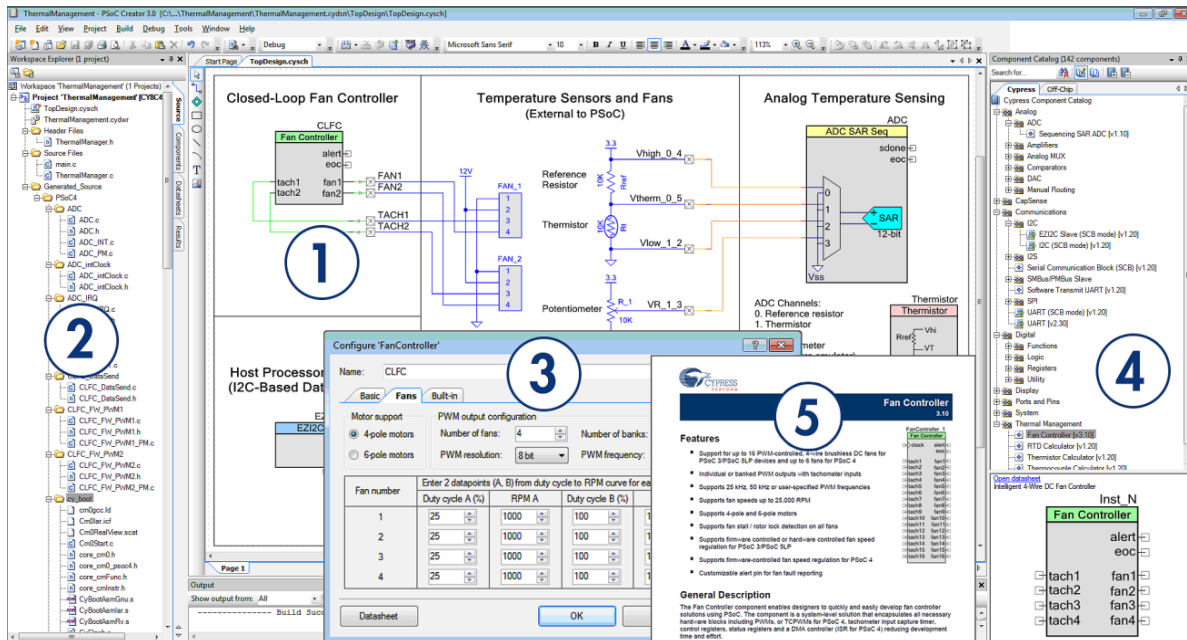
- **Overview:** [PSoC Portfolio](#), [PSoC Roadmap](#)
- **Product Selectors:** [PSoC 1](#), [PSoC 3](#), [PSoC 4](#), or [PSoC 5LP](#). In addition, [PSoC Creator](#) includes a device selection tool.
- **Datasheets** describe and provide electrical specifications for the PSoC 3, PSoC 4, and PSoC 5LP device families.
- **CapSense Design Guides:** Learn how to design capacitive touch-sensing applications with the PSoC 3, PSoC 4, and PSoC 5LP families of devices.
- **Application Notes** and **Code Examples** cover a broad range of topics, from basic to advanced level. Many of the application notes include code examples.
- **Technical Reference Manuals (TRM)** provide detailed descriptions of the architecture and registers
- in each of the PSoC 3, PSoC 4, and PSoC 5LP device families.
- **PSoC Training Videos:** These videos provide step-by-step instructions on getting started building complex designs with PSoC.
  - **Development Kits:**
    - [CY8CKIT-041](#) PSoC 4 S-Series Pioneer kit is easy-to-use and inexpensive development platform. This kit include connectors for Arduino™ compatible shields and Digilent® Pmod™ daughter cards.
    - [CY8CKIT-145](#) is a very low-cost prototyping platform for evaluating PSoC 4 S-Series devices.
    - The [MiniProg3](#) device provides an interface for flash programming and debug

## PSoC Creator

PSoC Creator is a free Windows-based Integrated Design Environment (IDE). It enables concurrent hardware and firmware design of systems based on PSoC 3, PSoC 4, and PSoC 5LP. See Figure 12 – with PSoC Creator, you can:

1. Drag and drop Components to build your hardware system design in the main design workspace
2. Codesign your application firmware with the PSoC hardware
3. Configure Components using configuration tools
4. Explore the library of 100+ Components
5. Review Component datasheets

Figure 12. PSoC Creator Features



## Document History

Document Title: CE210488 - LP CapSense® Buttons

Document Number: 002-10488

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5269057	SSHH/SRDS	11/18/2016	New code example

## 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>
Internet of Things	<a href="http://cypress.com/iot">cypress.com/iot</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](#)

### 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	Phone	: 408-943-2600
198 Champion Court	Fax	: 408-943-4730
San Jose, CA 95134-1709	Website	: <a href="http://www.cypress.com">www.cypress.com</a>

© Cypress Semiconductor Corporation, 2016. 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.