**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.

# CE224821 – PSoC 4 CapSense Touchpad with Gestures

## Objective

This code example demonstrates gesture detection in PSoC® 4 with a CapSense® touchpad using the PSoC Creator™ CapSense Component.

## Requirements

**Tool:** PSoC Creator 4.3

**Programming Language:** C (Arm® GCC 5.4.1)

**Associated Parts:** All PSoC 4 family devices that can support a CapSense Component with gesture detection

**Related Hardware:** CY8CKIT-041-41XX 4100S, CY8CKIT-041-40XX 4000S

## Overview

CapSense touchpads are used for controls requiring gradual adjustments. With the gesture detection capability, CapSense touchpads can detect a touch, position movement, and gesture. This allows the implementation of advanced user interfaces based on capacitive sensing.

This code example includes two projects that demonstrate the implementation of gesture detection on a CapSense touchpad using the CapSense Component. The first uses Self-Capacitance (CSD), while the second project uses Mutual-Capacitance (CSX). For more information, see Related Documents.

To demonstrate the gesture functionality, PSoC devices control three LEDs whose brightness is controlled based on the position detected on the touchpad. These are turned ON or OFF depending on the detected gestures. The CapSense tuner can also be used for real-time tuning and gesture detection.

## Hardware Setup

No hardware setup required.

## Software Setup

No software setup required.

## Operation

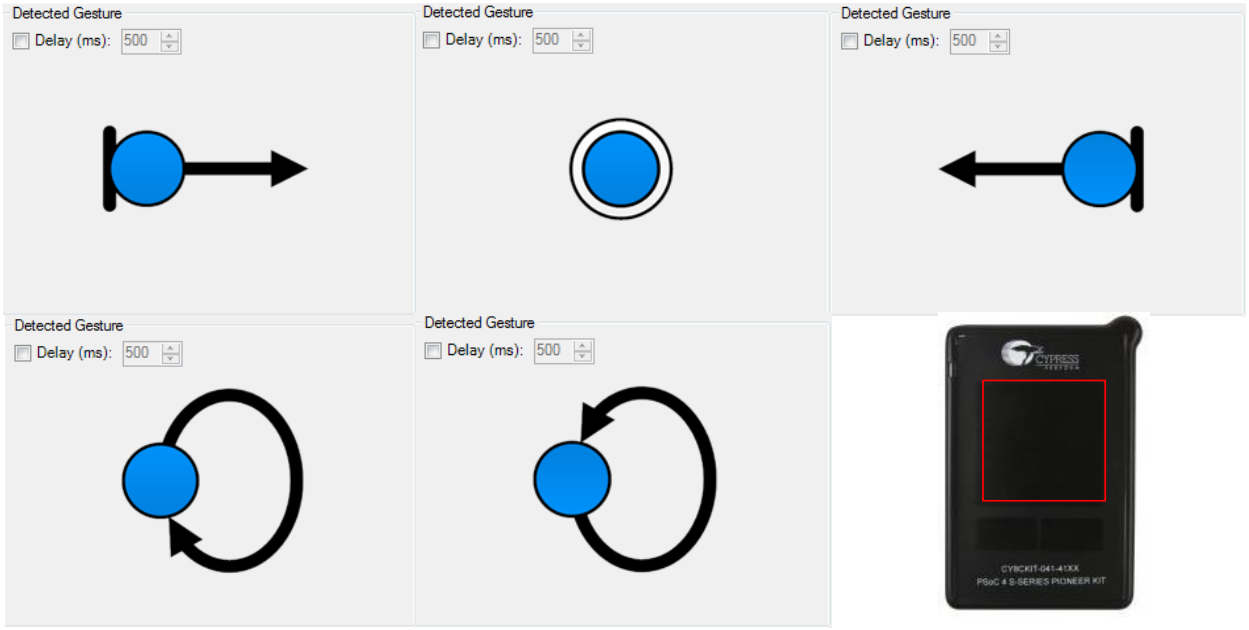
1. Connect CY8KIT-041-41XX to your computer using a USB cable. By default, the projects are configured for CY8KIT-041-41XX. To use the CY8CKIT-041-40XX, see Reusing this Example first.
2. Build the project and program it into the PSoC 4 device. Choose **Debug** > **Program**. For more information on device programming, see *PSoC Creator Help.*
3. Tap on the touchpad once and confirm that the blue and green LEDs turn ON.
4. Slide your finger slowly on the touchpad. Confirm that when your finger slides up and down, the green LED changes brightness. Confirm that when you slide your finger left and right, the blue LED changes brightness.
5. Place your finger on the left edge of the touchpad and swipe right. Confirm that the blue LED turns OFF; doing this again turns the LED back ON.
6. Place your finger on the right edge of the touchpad and swipe left. Confirm that the green LED turns OFF; doing this again turns the LED back ON.
7. Rotate your finger clockwise on the touchpad in a circular motion. Confirm that the red LED turns ON.
8. Rotate your finger counterclockwise on the touchpad in a circular motion. Confirm that the red LED turns OFF.

9. Right-click on the CapSense Component and select **Launch Tuner**. Click **Connect**, select **I2C**, and then click **Start**. Ensure that the data rate is set to **1 MHz**. Go to the **Gesture View** tab and select **Synchronized**. As each gesture is performed, confirm that the gesture shows up under the **Detected Gesture** portion of the page. For more information, check the CapSense datasheet under Related Documents.

## Design and Implementation

Each gesture is captured and is used to control different LEDs. The tuner also shows different gestures that the slider has detected and shows a picture on screen as shown in Figure 1. All CapSense gesture settings can be manipulated as seen in Components and Settings.

Figure 1. Edge Swipe Right, Click, Edge Swipe Left, Rotate Clockwise, Rotate Counterclockwise, Touchpad Area



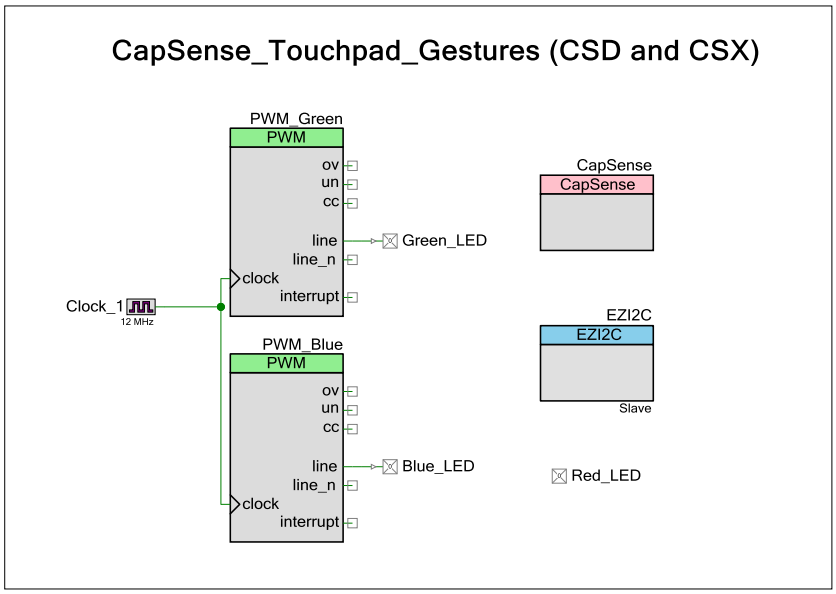

### CapSense Touchpad and Gestures

In the CapSense_Touchpad_Gestures_CSD and CSX examples, the following functions are performed:

1. Initialize and start all hardware.
2. Link the communication buffer for EZI2C to the CapSense data structure.
3. Set up a timestamp method to control gestures.
4. Initial scan of all CapSense widgets.
5. Check whether the scan was completed, and then process all data.
6. Store the gesture and use it in a switch statement to control the LEDs.
7. Send all data to the tuner.
8. Scan the CapSense widget and return to Step 5.

Figure 2 shows the top-design of the CSX and CSD PSoC Creator project:

Figure 2. CapSense_Touchpad_Gestures (CSD and CSX) Top Design Schematic




## Components and Settings

Table 1 lists the PSoC Creator Components used in this example, how they are used in the design, and the non-default settings required so they function as intended.

Table 1. PSoC Creator Components

| Component | Instance Name | Purpose | Non-default Settings |
|---|---|---|---|
| CapSense | CapSense | Gather and process all data from the selected sensors | **CSD** |
| | | | For Touchpad settings, see Figure 3.<br>For CSD Settings, see Figure 4.<br>For Widget Details, see Figure 5. |
| | | | **CSX** |
| | | | For Touchpad settings, see Figure 3, but change the Sense Mode to CSX.<br>For CSX settings, see Figure 6.<br>For Widget Details, see Figure 7. |
| | | | **CSD and CSX** have the same gestures setup.<br>For Click gesture settings, see Figure 8.<br>For Edge Swipe gesture settings, see Figure 9.<br>The Rotate gesture has no setting changes. |
| EZI2C | EZI2C | Transmits data from the kit to the tuner | Change the Data rate to **1000 (kbps)** and the Sub-address size to **16**. |
| PWM | PWM_Green<br>PWM_Blue | Controls the duty cycle of the green and blue LEDs | Both PWMs have the same hardware setup.<br>Under the PWM tab, deselect all interrupts, change the Period to **10001**, and the Compare value to **10001**. |
| Design Wide Resource | Clocks | Settings for all clocks | Click on IMO and change the Frequency to **48 MHz.** |

Figure 3. CSD CapSense Touchpad Settings

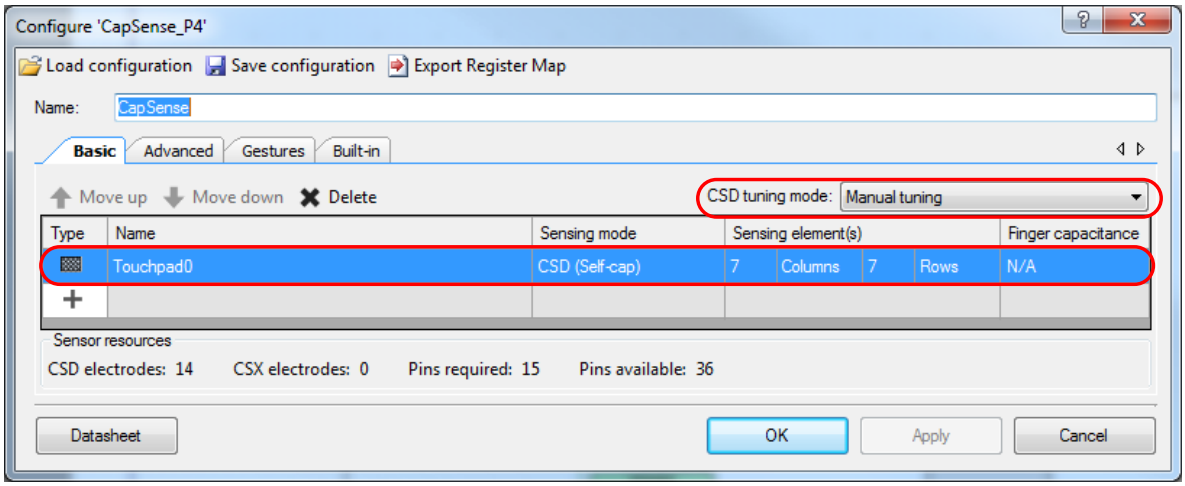
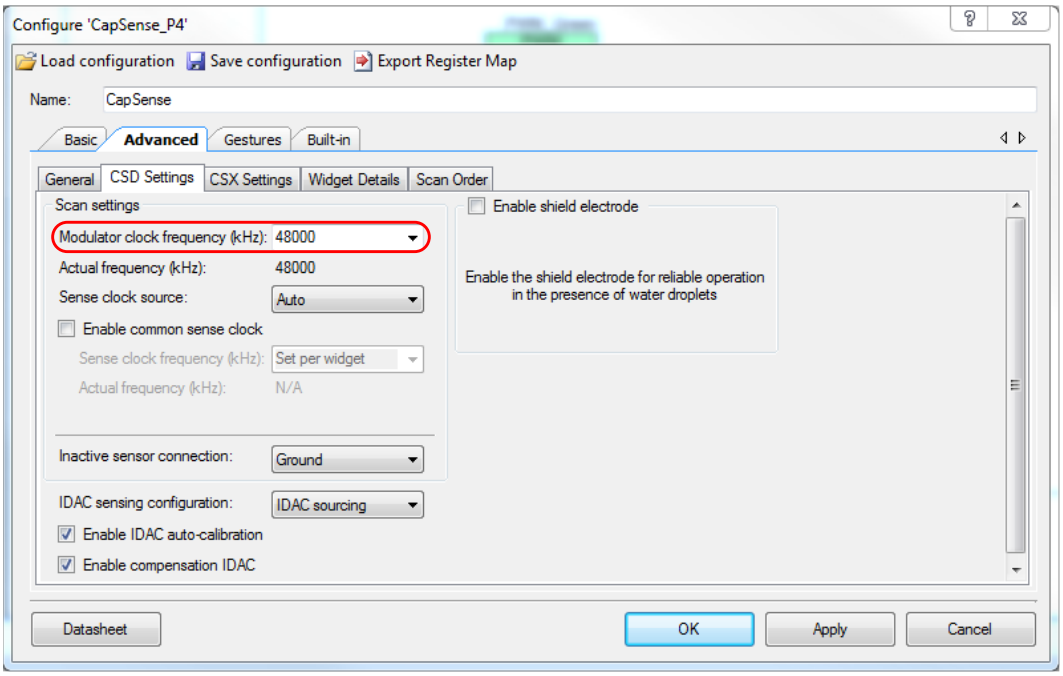



Figure 4. CSD Settings




**Note:** The Modulator Clock Frequency can be changed to 48,000 kHz only after the IMO clock is changed to 48 MHz. Under Design Wide Resources, click **Clocks** > **System IMO** > Select the drop down on the IMO Clock > **48 MHz**.

Figure 5. Widget Details

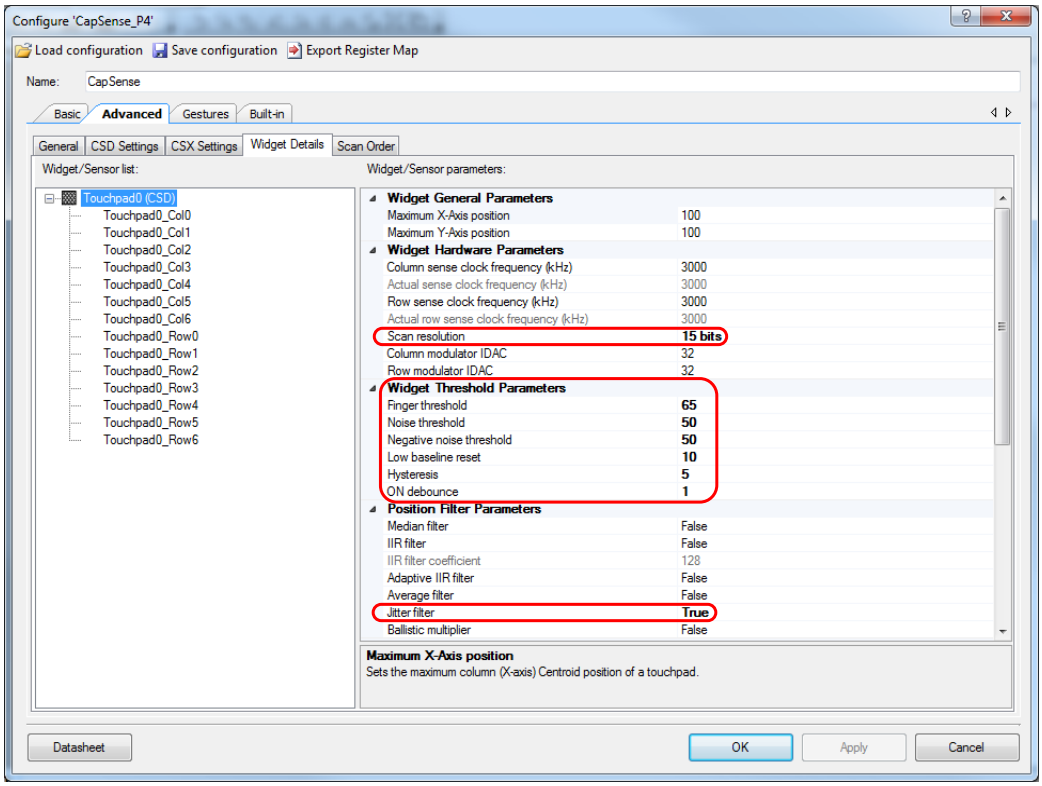
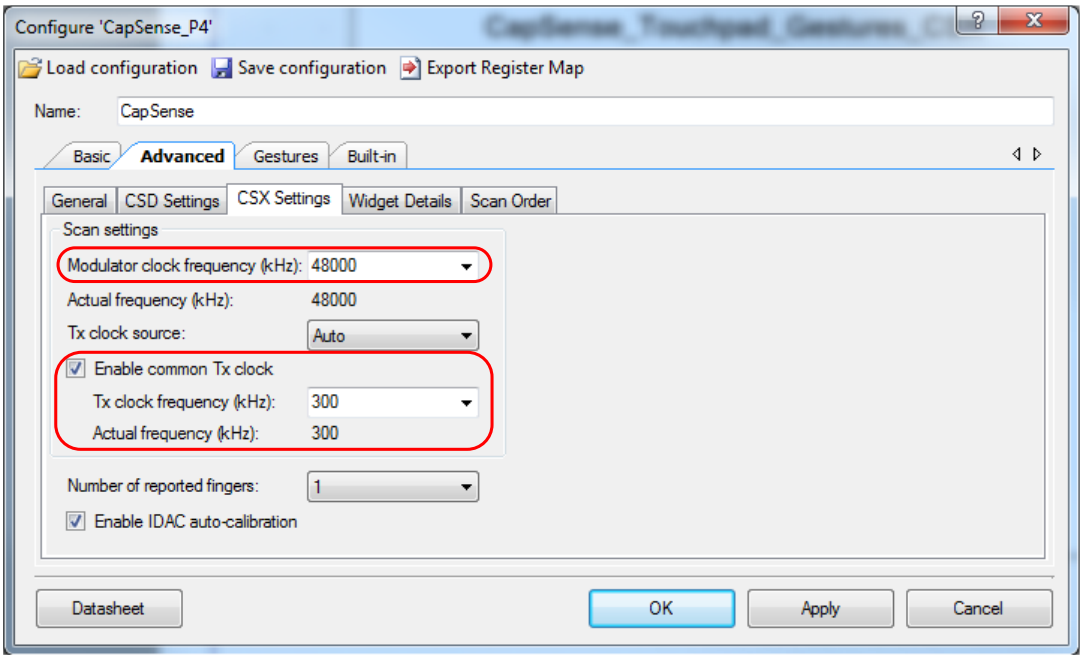



Figure 6. CSX Settings




**Note:** The Modulator Clock Frequency can be changed to 48,000 kHz only after the IMO clock is changed to 48 MHz. Under Design Wide Resources, click **Clocks** > **System IMO** > Select the drop down on the IMO Clock > **48 MHz**.

Figure 7. Widget Details

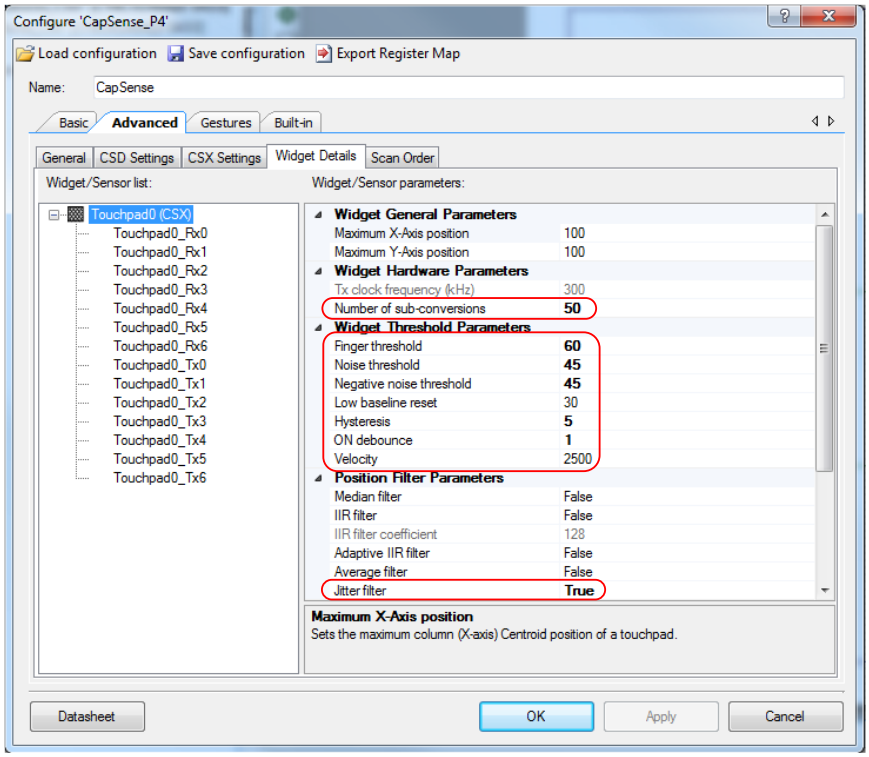



Figure 8. Click Gesture Settings

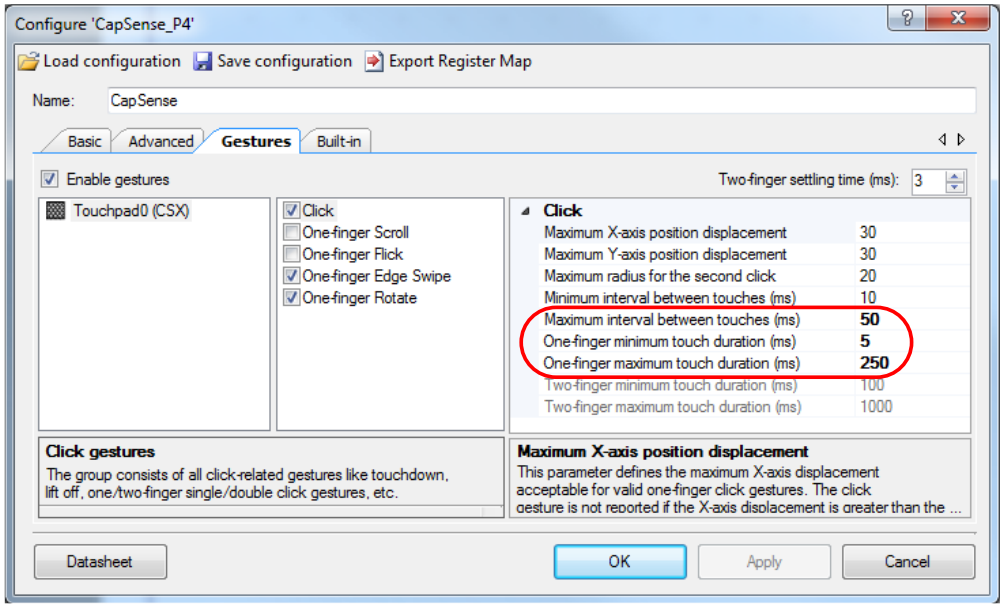
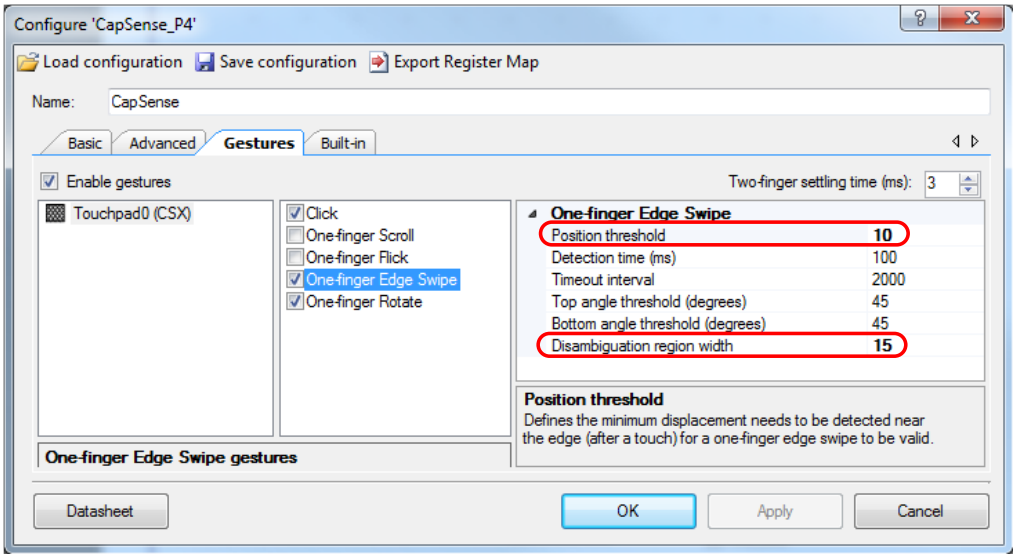


Figure 9. Edge Swipe Gesture Settings




For information on the hardware resources used by a Component, see the Component datasheet.

## Reusing this Example

This code example can be used with all PSoC 4 family devices that can support a CapSense Component with gesture detection. By default, the projects are set up to work with CY8KIT-041-41XX (MPN CY8C4146AZI-S433). If CY8CKIT-041-40XX (MPN CY8C4045AZI-S413) is used, follow the directions below. Ensure the following before porting this code example:

1. One of the two kits listed is used.
2. All the pins are unlocked in the Design Wide Resources.

To port the code to a new device, in PSoC Creator, select **Project** > **Device Selector** and change to the target device.

Before porting this example to another device, note the following:

1. Not all PSoC 4 devices have CapSense Components with gesture detection.
2. Pinouts change from device to device. Some pins may need to be moved. See the **Pin Layout** tab in PSoC Creator

In some cases, a resource used by a code example (for example, a Universal Digital Block) is not supported on another device. In that case, the example does not work. If you build the code targeted at such a device, errors occur. See the device datasheet for information on what a particular device supports.

## Related Documents

For a comprehensive list of PSoC 3, PSoC 4, and PSoC 5LP resources, see KBA86521 in the Cypress community.

| | |
|---|---|
| **Application Notes** | |
| AN79953 – Getting Started with PSoC® 4 | Describes PSoC 4 devices and how to build your first PSoC Creator project |
| AN85951 – PSoC 4 and PSoC 6 MCU CapSense Design Guide | Describes how to tune and use the CapSense Component |
| AN64846 – Getting Started with CapSense | Describes how to set up CapSense for beginners |
| **Code Examples** | |
| CE224820 – CapSense Slider and Gestures | Shows how to use Gestures on a slider |
| **PSoC Creator Component Datasheets** | |
| CapSense | CapSense Component datasheet for more information |
| TCPWM | TCPWM Component datasheet for more information |
| **Device Documentation** | |
| PSoC 4 Datasheets | PSoC 4 Technical Reference Manuals |
| **Development Kit Documentation** | |
| CY8CKIT-041-41XX PSoC 4100S CapSense Pioneer Kit | |
| CY8CKIT-041-40XX PSoC 4000S Pioneer Kit | |
| PSoC 4 Kits | |
| **Tool Documentation** | |
| PSoC Creator | Look in the **Downloads** tab for Quick Start and User Guides |

# Document History

Document Title: CE224821 – PSoC 4 CapSense Touchpad with Gestures

Document Number: 002-24821

| Revision | ECN | Submission Date | Description of Change |
|---|---|---|---|
| ** | 6385725 | 05/06/2019 | New code example |
| *A | 6951387 | 08/26/2020 | Minor updates to links.<br>Update documentation to show CY8KIT-041-41XX as the default device. |

## 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 | cypress.com/arm |
| Automotive | cypress.com/automotive |
| Clocks & Buffers | cypress.com/clocks |
| Interface | cypress.com/interface |
| Internet of Things | cypress.com/iot |
| Memory | cypress.com/memory |
| Microcontrollers | cypress.com/mcu |
| PSoC | cypress.com/psoc |
| Power Management ICs | cypress.com/pmic |
| Touch Sensing | cypress.com/touch |
| USB Controllers | cypress.com/usb |
| Wireless Connectivity | cypress.com/wireless |

## PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

## Cypress Developer Community

Community | Projects | Videos | Blogs | Training | Components

## Technical Support

cypress.com/support



Cypress Semiconductor
An Infineon Technologies Company
198 Champion Court
San Jose, CA 95134-1709