

## CY8CKIT-031

# PSoC<sup>®</sup> CapSense<sup>®</sup> Expansion Board Kit Guide

Doc. # 001-66474 Rev. \*D

Cypress Semiconductor 198 Champion Court San Jose, CA 95134-1709 Phone (USA): 800.858.1810 Phone (Intnl): 408.943.2600 http://www.cypress.com



#### Copyrights

© Cypress Semiconductor Corporation, 2011-2012. 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.

Any 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 MATE-RIAL, 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.

PSoC<sup>®</sup> and CapSense<sup>®</sup> are registered trademarks and PSoC Creator<sup>™</sup> and Programmable System-on-Chip are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

#### Flash Code Protection

Cypress products meet the specifications contained in their particular Cypress PSoC Data Sheets. Cypress believes that its family of PSoC products is one of the most secure families of its kind on the market today, regardless of how they are used. There may be methods, unknown to Cypress, that can breach the code protection features. Any of these methods, to our knowledge, would be dishonest and possibly illegal. Neither Cypress nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as 'unbreakable'.

Cypress is willing to work with the customer who is concerned about the integrity of their code. Code protection is constantly evolving. We at Cypress are committed to continuously improving the code protection features of our products.

# Contents



1.	Introdu	ction	3
	1.1	Kit Contents	3
	1.2	PSoC Creator	3
	1.3	Getting Started	4
	1.4	Additional Resources	4
		1.4.1 Beginner Resources	4
		1.4.2 Engineers Looking for More	4
		1.4.3 Learning from Peers	4
	1.5	Document History	5
	1.6	Documentation Conventions	5
2.	Installa	tion	7
	2.1	CD Installation	7
	2.2	Hardware	8
	2.3	Software	8
3.	Kit Ope	ration	9
4.	Code E	xamples	11
4.	Code E 4.1	xamples Code Example 1: BMM_USB	<b>11</b> .11
4.	<b>Code E</b> 4.1	xamples Code Example 1: BMM_USB 4.1.1 Project Description	<b>11</b> .11 .11
4.	<b>Code E</b> 4.1	xamples         Code Example 1: BMM_USB         4.1.1       Project Description         4.1.2       Hardware Connections	<b>11</b> .11 .11 .12
4.	<b>Code E</b> 4.1	xamples         Code Example 1: BMM_USB         4.1.1 Project Description         4.1.2 Hardware Connections         4.1.3 Verify Output	<b>11</b> .11 .11 .12 .13
4.	<b>Code E</b> 4.1 4.2	xamples         Code Example 1: BMM_USB         4.1.1 Project Description         4.1.2 Hardware Connections         4.1.3 Verify Output         Code Example 2: SLM_USB	<b>11</b> .11 .12 .13 .13
4.	<b>Code E</b> 4.1 4.2	xamples         Code Example 1: BMM_USB         4.1.1 Project Description         4.1.2 Hardware Connections         4.1.3 Verify Output         Code Example 2: SLM_USB         4.2.1 Project Description	<b>11</b> .11 .12 .13 .13 .13
4.	<b>Code E</b> 4.1 4.2	xamples         Code Example 1: BMM_USB         4.1.1 Project Description         4.1.2 Hardware Connections         4.1.3 Verify Output         Code Example 2: SLM_USB         4.2.1 Project Description         4.2.2 Hardware Connections	<b>11</b> .11 .12 .13 .13 .13 .13
4.	<b>Code E</b> 4.1 4.2	xamples         Code Example 1: BMM_USB         4.1.1 Project Description         4.1.2 Hardware Connections         4.1.3 Verify Output         Code Example 2: SLM_USB         4.2.1 Project Description         4.2.2 Hardware Connections         4.2.3 Verify Output	<b>11</b> .11 .12 .13 .13 .13 .13 .14
4.	<b>Code E</b> 4.1 4.2 4.3	xamples         Code Example 1: BMM_USB         4.1.1 Project Description         4.1.2 Hardware Connections         4.1.3 Verify Output         Code Example 2: SLM_USB         4.2.1 Project Description         4.2.2 Hardware Connections         4.2.3 Verify Output         Code Example 3: BMM_I2C	<b>11</b> .11 .12 .13 .13 .13 .14 .14
4.	<b>Code E</b> 4.1 4.2 4.3	xamples         Code Example 1: BMM_USB         4.1.1 Project Description         4.1.2 Hardware Connections         4.1.3 Verify Output         Code Example 2: SLM_USB         4.2.1 Project Description         4.2.2 Hardware Connections         4.2.3 Verify Output         Code Example 3: BMM_I2C         4.3.1 Project Description	<b>11</b> .11 .12 .13 .13 .13 .13 .14 .14 .15 .15
4.	<b>Code E</b> 4.1 4.2 4.3	xamples         Code Example 1: BMM_USB         4.1.1 Project Description         4.1.2 Hardware Connections         4.1.3 Verify Output         Code Example 2: SLM_USB         4.2.1 Project Description         4.2.2 Hardware Connections         4.2.3 Verify Output         Code Example 3: BMM_I2C         4.3.1 Project Description         4.3.2 Hardware Connections	<b>11</b> .11 .12 .13 .13 .13 .14 .14 .15 .15
4.	<b>Code E</b> 4.1 4.2 4.3	xamples         Code Example 1: BMM_USB         4.1.1 Project Description         4.1.2 Hardware Connections         4.1.3 Verify Output         Code Example 2: SLM_USB         4.2.1 Project Description         4.2.2 Hardware Connections         4.2.3 Verify Output         Code Example 3: BMM_I2C         4.3.1 Project Description         4.3.2 Hardware Connections         4.3.3 Verify Output	<b>11</b> .11 .12 .13 .13 .13 .13 .14 .15 .15 .15 .16
4.	<b>Code E</b> 4.1 4.2 4.3	xamples         Code Example 1: BMM_USB         4.1.1 Project Description         4.1.2 Hardware Connections         4.1.3 Verify Output         Code Example 2: SLM_USB         4.2.1 Project Description         4.2.2 Hardware Connections         4.2.3 Verify Output         Code Example 3: BMM_I2C         4.3.1 Project Description         4.3.2 Hardware Connections         4.3.3 Verify Output         Code Example 4: SLM_I2C	<b>11</b> .11 .12 .13 .13 .13 .14 .15 .15 .15 .16 .21
4.	Code E 4.1 4.2 4.3 4.4	xamplesCode Example 1: BMM_USB4.1.1 Project Description4.1.2 Hardware Connections4.1.3 Verify OutputCode Example 2: SLM_USB4.2.1 Project Description4.2.2 Hardware Connections4.2.3 Verify OutputCode Example 3: BMM_I2C4.3.1 Project Description4.3.2 Hardware Connections4.3.3 Verify OutputCode Example 4: SLM_I2C4.4.1 Project Description	<b>11</b> .11 .12 .13 .13 .13 .14 .15 .15 .15 .16 .21
4.	Code E 4.1 4.2 4.3 4.4	xamplesCode Example 1: BMM_USB4.1.1 Project Description4.1.2 Hardware Connections4.1.3 Verify OutputCode Example 2: SLM_USB4.2.1 Project Description4.2.2 Hardware Connections4.2.3 Verify OutputCode Example 3: BMM_I2C4.3.1 Project Description4.3.2 Hardware Connections4.3.3 Verify OutputCode Example 4: SLM_I2C4.4.1 Project Description4.4.2 Hardware Connections	<b>11</b> .11 .12 .13 .13 .13 .14 .15 .15 .16 .21 .21



Α.	Append	ix	25
	A.1	Schematic	25
	A.2	Board Layout	26
		A.2.1 PDC-09801 Top	26
		A.2.2 PDC-09801 Bottom	26
	A.3	BOM	27
	A.4	Board Files	27



Thank you for your interest in the CY8CKIT-031 PSoC<sup>®</sup> CapSense<sup>®</sup> Expansion Board Kit. The PSoC CapSense Expansion Board Kit interfaces any of the CY3280 Universal CapSense Module boards with the CY8CKIT-001 PSoC Development Kit or CY8CKIT-030 PSoC 3 Development Kit. This kit enables you to develop CapSense solutions with the CY3280 Universal CapSense Module boards. The CapSense component in PSoC Creator<sup>™</sup> allows you to develop CapSense solutions with ease. The code examples provided along with this kit give sample solutions.

The PSoC CapSense Expansion Board is based on the PSoC 3 and PSoC 5 family of devices. PSoC 3 and PSoC 5 are based on Programmable System-on-Chip<sup>™</sup> platform for 8-, 16-, and 32-bit applications. It combines precision analog and digital logic with a high-performance 8051 single cycle per instruction pipelined processor achieving ten times the performance of previous 8051 processors. With the PSoC, you can create the exact combination of peripherals and integrated proprietary IP to meet your application needs.

## 1.1 Kit Contents

The CY8CKIT-031 PSoC CapSense Expansion Board Kit includes:

- CY8CKIT-031 PSoC CapSense Expansion Board
- CY3280-BMM Universal CapSense Matrix Button Module Kit
- CY3280-SLM Universal CapSense Linear Slider Module Kit
- Two 1.5 mm overlay and one 3 mm overlay
- Quick Start Guide
- Resource CD

## 1.2 PSoC Creator

Cypress's PSoC Creator software is a state-of-the-art, easy-to-use integrated development environment (IDE) that introduces a hardware and software design environment based on classic schematic entry and revolutionary embedded design methodology.

With PSoC Creator, you can:

- Automatically place and route select components and integrate simple glue logic normally located in discrete muxes.
- Trade off hardware and software design considerations allowing you to focus on what matters and getting to market faster.

PSoC Creator also enables you to tap into an entire tools ecosystem with integrated compiler tool chains, RTOS solutions, and production programmers to support both PSoC 3 and PSoC 5.



## 1.3 Getting Started

To get started, go to Kit Operation chapter on page 9 for a description of the kit operation. This chapter explains how the CapSense expansion board kit connects to a development kit and CY3280 module boards. The Code Examples chapter on page 11 explains code examples provided with the kit. The Appendix chapter on page 25 provides the schematics and bill of materials (BOM) associated with the CapSense Expansion Board Kit.

## 1.4 Additional Resources

Visit http://www.cypress.com/go/training for additional learning resources in the form of datasheets, technical reference manual, and application notes.

### 1.4.1 Beginner Resources

AN54181 - PSoC 3 - Getting Started with a PSoC 3 Design Project PSoC Creator Training

## 1.4.2 Engineers Looking for More

AN54460 - PSoC 3 and PSoC 5 Interrupts

AN52705 - PSoC 3 and PSoC 5 - Getting Started with DMA

AN52701 - PSoC 3 - How to Enable CAN Bus Communication

AN54439 - PSoC 3 and PSoC 5 External Oscillator

AN52927 - PSoC 3: Segment LCD Direct Drive

Cypress continually strives to provide the best support. Click here to view a growing list of application notes for PSoC 3 and PSoC 5.

#### 1.4.3 Learning from Peers

Cypress Developer Community Forums



## 1.5 Document History

Revision	PDF Creation Date	Origin of Change	Description of Change	
**	03/09/11	PVKV	Initial version of kit guide	
*A	03/30/11	PVKV	Added notes in Code Examples chapter on page 11	
*В	06/22/11	SSUT	Updated Verify Output on page 16	
*C	12/16/11	SASH	Updated PSoC Creator version	
*D	04/23/12	SASH	Updated the Additional Resources on page 4	

## **1.6 Documentation Conventions**

Table 1-1.	Document	Conventions <sup>•</sup>	for	Guides
------------	----------	--------------------------	-----	--------

Convention	Usage
Courier New	Displays file locations, user entered text, and source code: C:\cd\icc\
Italics	Displays file names and reference documentation: Read about the <i>sourcefile.hex</i> file in the <i>PSoC Designer User Guide</i> .
[Bracketed, Bold]	Displays keyboard commands in procedures: [ <b>Enter</b> ] or [ <b>Ctrl</b> ] [ <b>C</b> ]
File > Open	Represents menu paths: File > Open > New Project
Bold	Displays commands, menu paths, and icon names in procedures: Click the <b>File</b> icon and then click <b>Open</b> .
Times New Roman	Displays an equation: 2 + 2 = 4
Text in gray boxes	Describes cautions or unique functionality of the product.

Introduction







## 2.1 CD Installation

To install the CY8CKIT-031 PSoC CapSense Expansion Board Kit software, insert the kit CD into the CD drive of your computer. The CD is designed to auto-run and the PSoC CapSense Expansion Board Kit menu appears.



Note If auto-run does not execute, double-click AutoRun on the root directory of the CD.



After the installation is complete, the kit contents are available in the following location: C:\Program Files\Cypress\PSoC CapSense EBK\2.0\ Installation



The setup installs the following software:

- PSoC Creator
- PSoC Programmer
- Kit Documentation
  - Quick Start Guide
  - □ Kit Guide
  - Known Problems and Solutions
  - Release Notes
- Firmware
  - Code Examples
- Hardware
  - Schematic
  - Layout
  - □ BOM

## 2.2 Hardware

**WARNING** Static discharges from the human body can easily reach 20,000 volts. This can damage the PSoC 3 device on the development kit. Ensure that any static is discharged before touching the hardware.

- Power off the development kit before making any connections
- Connect the PSoC CapSense Expansion Board Kit to the development kit being used
- Connect the CY3280-Universal CapSense Module Board to the PSoC CapSense Expansion Board
- Power the development kit

## 2.3 Software

When installing the PSoC CapSense Expansion Board Kit, the installer checks if the prerequisites, PSoC Creator, PSoC Programmer, Windows Installer, .NET, Acrobat Reader, and KEIL Complier, are installed in your computer. If these applications are not installed, then the installer prompts you to download and install them.





The following figures show the CY8CKIT-031 PSoC CapSense Expansion Board Kit, CY3280 CapSense Matrix Button Module, and CY3280 CapSense Linear Slider Module.

Figure 3-1. CY8CKIT-031 PSoC CapSense Expansion Board Kit



1x5 Pin Connector for I2C Interface

#### Figure 3-2. CapSense Matrix Button Module







The PSoC CapSense Expansion Board Kit connects to the development kit using a 2×20-pin connector. It connects to the CY3280 Universal CapSense Module Boards using 2×22-pin connector.



Figure 3-4. PSoC CapSense Expansion Board Kit Interface with PSoC Development Kit and Universal CapSense Matrix Button Module Kit



Figure 3-5. PSoC CapSense Expansion Board Kit Interface with PSoC 3 Development Kit and Universal CapSense Matrix Button Module Kit



The 5-pin connector provides the  $I^2C$  interface.

Figure 3-6. I2C Connector



**Note** The PSoC CapSense Expansion Board Kit can use only Port A of the CY8CKIT-001 PSoC Development Kit. Other ports - Port A', Port B, and Port C cannot be used with this kit.

# 4. Code Examples



Four code examples are provided with the kit.

- Code Example 1: BMM\_USB
- Code Example 2: SLM\_USB
- Code Example 3: BMM\_I2C (Works only with MiniProg3)
- Code Example 4: SLM\_I2C (Works only with MiniProg3)

The code examples work with the following development kits:

- CY8CKIT-001 PSoC Development Kit
- CY8CKIT-030 PSoC 3 Development Kit

The PSoC Development Kit examples are named with the suffix \_*KIT-001*; the PSoC 3 Development Kit examples are named with the suffix \_*KIT-030*. Use the appropriate examples according to development kit being used.

All code examples are available in the directory C:\Program Files\Cypress\PSoC CapSense EBK\2.0\Firmware.

## 4.1 Code Example 1: BMM\_USB

#### 4.1.1 Project Description

This example shows the CapSense Matrix Button Module (BMM) interfacing with the development kit (CY8CKIT-001 or CY8CKIT-030) through the PSoC CapSense Expansion Board Kit. The example demonstrates the CapSense button matrix operation. When any button is touched, the corresponding row and column LEDs turn on. This example also shows the calculator application in the PC with the CapSense BMM as a keypad.

The BMM module has four row sensors and four column sensors. Touching a button activates the corresponding row and column sensors. The example uses the auto-tuning feature, which sets all CapSense parameters to best values automatically. The USBFS component is configured as a HID keyboard. The buttons on the BMM are mapped to keys on the calculator as follows:

	Col 0	Col 1	Col 2	Col3
Row 0	0	1	2	3
Row 1	4	5	6	7
Row 2	8	9		=
Row 3	+	_	*	/

Before entering the main loop, the code waits for two seconds for USB enumeration. An LED named Enumerate\_LED is tuned on to indicate that wait time for USB enumeration is over. The main loop continuously scans all the sensors. If a buttons is touched, the corresponding row LED and column LED are turned on and the corresponding key code is sent to the PC via USB.



The Sensitivity parameter in the CapSense\_CSD component configuration is set to '4' and the example is tested to work without any overlay. When an overlay of some thickness is used, the sensitivity parameter should be changed, as shown in Figure 4-1. The sensitivity parameter indicates the finger capacitance, which depends on the button area, overlay thickness, and dielectric constant of overlay material. Options for sensitivity are 1 to 4. Setting this to '1', indicates that a button touch adds a 0.1-pF capacitance. Setting this to '4' indicates that a button touch adds a 0.4-pF capacitance.

Name         CapSense           General         Widgets Coofig         Scan Order         Advanced         Tune Helper         Built-in                Promote	4 Þ	s CapSense
General     Widgets Config     Scan Order     Advanced     Tune Helper     Built-in       Image: Promote <ul> <li>Demote</li> <li>Demote</li> <li>Scan Skit</li> <li>Chi Di Sensor</li> <li>Małkitkutoriji Coll_MB</li> <li>Małkitkutoriji Coll_MB</li></ul>	4 Þ	
Promote         Demote           Scan Slat         Ch0 Sensor           0         MathbUtori0_Col0_MB           1         MathbUtori0_Col1_MB           2         MathbUtori0_Col2_MB           3         MathbUtori0_Col3_MB           4         MathbUtori0_Row0_MB           5         MathbUtori0_Row1_MB		General Widgets Config Scan Order Advanced Tune Helper Built-in
Scan Slot     Ch0 Sensor       0     MathoButton0_Col0_MB       1     MathoButton0_Col1_MB       2     MathoButton0_Col2_MB       3     MathoButton0_Col3_MB       4     MathoButton0_Row0_MB       5     MathoButton0_Row1_MB		romote + Demote
0         MathoButori0_Col0_MB           1         MathoButori0_Col1_MB           2         MathoButori0_Col2_MB           3         MathoButori0_Col3_MB           4         MathoButori0_Row0_MB           5         MathoButori0_Row1_MB		en Slot Ch0 Sensor
Mathburon0_Coll_M8           2         Mathburon0_Col2_M8           3         Mathburon0_Col3_M8           4         Mathburon0_Row0_M8           5         Mathburon0_Row1_M8		Mahodluton0_Col0M8
MathoButton0_Col2_M8           3         MathoButton0_Col3_M8           4         MathoButon0_Row0_M8           5         MathoButon0_Row1_M8		MatrixButton0_Col1MB
3 MathoButton0_Col3_MB 4 MathoButton0_Row0_M8 5 MathoButton0_Row1_M8		MatrixButton0_Col2MB
4 MahisButon0_Row0_MB 5 MahisButon0_Row1_MB		MatrixButton0_Col3MB
5 MatrixButton0_Row1MB		MatrixButton0_Row0M8
		MatrixButton0_Row1MB
6 MatrixButton0_Row2MB		MatrixButton0_Row2MB
MativButton0_Row3MB		MatrixButton0_Row3MB
o realloculute_novewo 7 Mativ6ution0_Row3_M8		MatixBution0_Row3_MB

Figure 4-1. Sensitivity Parameter Setting

#### 4.1.2 Hardware Connections

For the PSoC Development Kit

- Connect the CapSense Matrix Button Module to connector J2 of the PSoC CapSense Expansion Board.
- Connect J1 of the PSoC CapSense Expansion Board to port A of the development kit.
- Connect J2 on the CapSense Matrix Button Module to short SHIELD and SHLD.
- Connect the USB cable from J9 on the development kit to the PC USB port.
- Connect P1[6] to LED1. This LED is named Enumerate\_LED and indicates that USB enumeration is complete.

For the PSoC 3 Development Kit

- Connect the CapSense Matrix Button Module to connector J2 of the PSoC CapSense Expansion Board.
- Connect J1 of the PSoC CapSense Expansion Board to port D of the development kit.
- Connect J2 on the CapSense Matrix Button Module to short SHIELD and SHLD.
- Connect jumpers J10 and J11 to position 2 and 3 to power the board at 5 V.
- Connect the USB cable from J2 on the development kit to the PC USB port.
- Pin P6[3] is used for the Enumerate\_LED, which is connected to LED4 on the board; therefore, an explicit connection is not required.
- Remove LCD from port P8. The LCD module adds parasitic capacitance and noise to CapSense because it shares the same pins of port D that are used for CapSense.
- If MiniProg3 is used, then disconnect it from J3 after programming. The programming port J3 shares the port D pins and adds the noise on CapSense if MiniProg3 is present.



## 4.1.3 Verify Output

Build and program the code example and reset the device. After device is reset, wait for Enumerate\_LED to turn on. The LED indicates that 2 seconds wait time for USB enumeration is over. Touch a button and see the corresponding row and column LEDs turn on. Open the calculator application in the PC, touch any button, the corresponding key is pressed on calculator. Perform different operations on the calculator using CapSense touch buttons.



Figure 4-2. CapSense Matrix Button Module Project

## 4.2 Code Example 2: SLM\_USB

## 4.2.1 Project Description

This example shows the CapSense Linear Slider Module (SLM) interfacing with the development kit (CY8CKIT-001 or CY8CKIT-030) through the PSoC CapSense Expansion Board. The example demonstrates the CapSense slider and button combination. Touch a button on the module turns on the corresponding LED; placing a finger on the slider, turns on the LED nearest to the finger position. The example also shows a media player application in the PC with buttons and slider on the CapSense SLM as media controls such as play/pause and volume control.

The SLM module has 10 element slider and five buttons. The module also has five LEDs. The finger position on the slider is indicated by turning on the nearest LED. The active button is indicated by turning on the corresponding LED. The project uses the auto-tuning feature, which sets all CapSense parameters to best values automatically. The USBFS user module is configured as a HID keyboard. The buttons and slider on the SLM are used as controls for the media player as follows.

- BTN0 Play/Pause
- BTN1 Stop
- BTN2 Mute/UnMute
- BTN3 Next Track
- **BTN4** Previous Track
- Slider Left/Right Volume UP/DOWN

Before entering the main loop, the code waits for two seconds for USB enumeration. An LED named Enumerate\_LED is turned on to indicate that wait time for USB enumeration is over. The main loop continuously scans all the sensors. LEDs are updated based on active buttons and finger position on the slider and key code is sent to the PC via USB.

The Sensitivity parameter in the CapSense\_CSD component configuration is set to '4' and the example is tested to work without any overlay. When overlay is used, the sensitivity parameter should be changed in the project, as shown in Figure 4-1 on page 12.

![](_page_15_Picture_1.jpeg)

## 4.2.2 Hardware Connections

For the PSoC Development Kit

- Connect the CapSense Linear Slider Module to connector J2 of the PSoC CapSense Expansion Board.
- Connect J1 of the PSoC CapSense Expansion Board Kit to port A of the development kit.
- Connect J2 on the CapSense Linear Slider Module to short SHIELD and SHLD.
- Connect jumpers J10 and J11 to position 1 and 2 to power the board at 3.3 V.
- Connect the USB cable from J9 on the development kit to the PC USB port.
- Connect P1[6] to LED1. This LED is named Enumerate\_LED and indicates that USB enumeration is complete.

For the PSoC 3 Development Kit

- Connect the CapSense Linear Slider Module to connector J2 of the PSoC CapSense Expansion Board.
- Connect J1 of the PSoC CapSense Expansion Board Kit to the port D of the development kit.
- Connect J2 on the CapSense Linear Slider Module to short SHIELD and SHLD.
- Connect the USB cable from J2 on the development kit to the PC USB port.
- Pin P6[3] is used for Enumerate\_LED, which is connected to LED4 on the board; therefore, an explicit connection is not required.
- Remove LCD from port P8. The LCD module adds parasitic capacitance and noise to CapSense because it shares the same pins of port D that are used for CapSense.
- If MiniProg3 is used, then disconnect it from J3 after programming. The programming port J3 shares the port D pins and adds the noise on CapSense if MiniProg3 is present.

## 4.2.3 Verify Output

Build and program the code example and reset the device. After the device is reset, wait for Enumerate\_LED to turn on. The LED indicates that 2 seconds wait time for USB enumeration is over. Touch a button and see the corresponding LED turn on. Place a finger on the slider and see the nearest LED turn on. Open the Windows Media Player 11 application in the PC. Perform different operations such as Play/Pause, Next, Previous, and Volume Control using CapSense Linear Slider Module touch controls.

Figure 4-3. CapSense Linear Slider Module Project

![](_page_15_Picture_21.jpeg)

![](_page_16_Picture_1.jpeg)

## 4.3 Code Example 3: BMM\_I2C

**Note** The project uses MiniProg3. It does not work without MiniProg3 because the code waits forever in the main loop if I2C communication is not set up. You can buy MiniProg3 through this link http://www.cypress.com/go/CY8CKIT-002.

### 4.3.1 Project Description

This code example shows the CapSense Matrix Button Module (BMM) with "Tuner" for monitoring of CapSense outputs. The CapSense outputs such as Rawcounts, Baseline, and Signal (Difference count) can be monitored on the "Tuner" GUI. The project uses the auto-tuning feature, which sets all CapSense parameters to best values automatically. The parameter settings can be monitored in the GUI but cannot be altered because they are set by auto-tuning. In the manual tuning method, parameter settings can be changed in the GUI and the resulting output can be seen.

The example also makes use of LEDs on the CapSense BMM board. Touching a button turns on the corresponding LED. The code uses tuner APIs. The tuner API CapSense\_TunerComm() is used in main loop to scan sensors, which also sends the CapSense variables RawCounts, Baseline, and Difference counts (Signal) to the PC GUI through I2C communication.

The Sensitivity parameter in the CapSense\_CSD component configuration is set to '4' and the example is tested to work without any overlay. When an overlay of some thickness is used, the sensitivity parameter should be changed in the project, as shown in Figure 4-1 on page 12.

### 4.3.2 Hardware Connections

- Connect the CapSense Matrix Button Module to connector J2 of the PSoC CapSense Expansion Board.
- For the PSoC Development Kit, connect J1 of the PSoC CapSense Expansion Board Kit to port A
  of the development kit.
- For the PSoC 3 Development Kit, connect J1 of the PSoC CapSense Expansion Board Kit to port D of the development kit. Remove LCD from port P8. LCD module adds parasitic capacitance and noise to CapSense because it shares the same pins of port D that are used for CapSense.
- Connect jumper J2 on the CapSense Matrix Button Slider Module to short SHIELD and SHLD.
- If MiniProg3 is used, then disconnect it from the programming port and PC. The MiniProg3 is used for I2C communication. Connect MiniProg3 to the J3 header on the PSoC CapSense Expansion Board Kit. Make sure that I2C pins SDA, SCL, and GND on the MiniProg3 are mapped to the corresponding I2C pins on the kit; see Figure 3-6 on page 10 for the connections. Note that when MiniProg3 is used for both I2C and programming, the MiniProg3 should be disconnected from PC every time the connection is changed.

![](_page_17_Picture_1.jpeg)

## 4.3.3 Verify Output

Build and program the code example. If MiniProg3 is used, disconnect it from the programming header and connect it to I2C connector J3 on the PSoC CapSense Expansion Board Kit to use it for communication with GUI. Reset the device.

#### **Tuner GUI Use**

1. To open the Tuner, right-click on the **CapSense\_CSD** component in PSoC Creator and click on **Launch Tuner.** 

![](_page_17_Picture_6.jpeg)

2. Click on the **Configuration** button to open the configuration window.

![](_page_17_Picture_8.jpeg)

![](_page_18_Picture_1.jpeg)

3. Set the I2C communication parameters same as that set in the EZI2C component.

Tuner Communication Setup			? 🛛
Ports	Port Configuration		
MiniProg3 SN:MiniP31014DD000112	I2C Voltage:	3.3V	*
	I2C Bus Speed	400kHz	*
	I2C Address:	4	
	Sub-address:	2-Bytes	*
Pool: Version: 2.5 Pw/Version: 2.95			
	ОК	Cano	el 🛛

4. Select any option for the I2C Voltage other than **External**. The external supply pin on the MiniProg3 VTARG connects to NC on the CY8CKIT-031 board. Therefore, the external supply option does not work with the CY8CKIT-030.

Tuner Communication Setup	? ×
Ports:	Port Configuration I2C Voltage: 3.3 V I2C Bus Speed: I2C Address: 2.5 V 3.3 V Sub-address: 50 V Sub-address: 50 V
Port Information FPGA Version: 1.14 PSoC Version: 2.5 FW Version: 2.95	
	OK Cancel

5. Click **OK** to apply the settings.

![](_page_19_Picture_1.jpeg)

L CapSense_CSD Tuner for 'CapSense', Tuning Method: Au	ito (SmartSense).	_ 🗆 🗡
File Debug Tools		🚺 Help
🗤 Start 🛄 Stop 🛛 🔯 Configuration 🛛 🔚 Enable Logging		
Tuning Graphing Logging		
🖻 Reset Widgets Layout	Max Va	ilue: 255
MatrixButton0		200
		- 200
		150
		100
		100
		- 50
		0
3	SNR 📕	
		Revert Parameters
0 1 2 3		A THE REPORT OF THE REPORT OF
	Analog	asw 3 🔶
	I dach Scans	ang ts_200uA
	∕~ OK	X Cancel
Speed: Desired Packet Size: 61		

6. To start the scanning and communication process, click Start.

- 7. Open the **Graphing** tab; it shows different CapSense results RawCounts, Baseline, and Difference count (signal) for each sensor.
- 8. Select the sensor parameter to observe, as shown in Figure 4-4. See the graph for different variables.
- 9. Touch the button and observe the increase in counts.

![](_page_20_Picture_0.jpeg)

![](_page_20_Figure_2.jpeg)

Figure 4-4. Graphs for Matrix Button Module Project

Touch any button and observe the corresponding LED turned on. Figure 4-4 shows the monitoring of RawCounts, Baseline, and Difference count (signal) for the Column 0 sensor.

9. Open the **Tuning** tab and select a sensor row or column. Different CapSense parameters are shown on the bottom-right. Because auto-tuning is used in this project, you cannot edit the settings. Auto-tuning automatically sets all the parameters. The GUI is used to monitor the CapSense variables RawCounts, Baseline, and Signal for all sensors. Touch the button 0 (Row 0 Col 0) to see the sensor response in the tuner window, as shown in Figure 4-5. Also, observe that the corresponding row and col LED on the module board turns on.

![](_page_21_Picture_1.jpeg)

	ug) ris
Start Configuration Enable Logging	
ning Graphing Logging	
Reset Widgets Layout	Max Value: 255
atrixB utton0	
	- 20
	- 15
	- 1
	- 5
2	o
	SNR 8.7383286
3	Pavert Parameter
0 1 2 3	Name: MatrixButton Bow Find 85
	Row Nois 42
	Row Hyst 21
	Dem Can 14 kits
	AnalogSw 3
	ScanSpel East

Figure 4-5. Tuner Window

![](_page_22_Picture_1.jpeg)

## 4.4 Code Example 4: SLM\_I2C

#### Notes

- The example uses MiniProg3. It does not work without MiniProg3 because the code waits forever in the main loop if I2C communication is not set up. You can buy MiniProg3 at http://www.cypress.com/go/CY8CKIT-002.
- The example uses Port E for the PSoC 3 Development Kit and includes an LCD to display the slider position.

#### 4.4.1 Project Description

This code example shows the CapSense Linear Slider Module (SLM) with "Tuner" for monitoring CapSense outputs. The CapSense outputs such as Rawcounts, Baseline, and Signal (Difference count) can be monitored on the "Tuner" GUI. The project uses the auto-tuning feature, which sets all CapSense parameters to best values automatically.

The project includes a character LCD for displaying the slider position. The slider position is displayed as a horizontal bar graph on the character LCD. The LEDs on the CapSense SLM board turn on when the corresponding button is touched. When a slider is touched, the nearest LED turns on. The code uses tuner APIs. The tuner API CapSense\_TunerComm() is used in the main loop to scan sensors, which also sends the CapSense variables RawCounts, Baseline, and Difference counts (Signal) to the PC GUI through I2C communication.

The Sensitivity parameter in the CapSense\_CSD component configuration is set to '4' and the example is tested to work without any overlay. When an overlay is used, the sensitivity parameter should be changed in the project, as shown in Figure 4-1 on page 12.

#### 4.4.2 Hardware Connections

**Note** The example uses **Port E** for the PSoC 3 Development Kit instead of port D as with the other examples. This is to accommodate the LCD, which uses the same pins of port D as the CapSense.

- Connect the CapSense Linear Slider Module to connector J2 of the PSoC CapSense Expansion Board.
- Connect jumper J2 on the CapSense Linear Slider Module to short SHIELD and SHLD.
- For the PSoC Development Kit
  - □ Connect J1 of the PSoC CapSense Expansion Board Kit to port A of the development kit.
  - □ Connect the LCD to P18 and place jumper J12 in the ON position to power on the LCD.
- For the PSoC 3 Development Kit
  - □ Connect J1 of the PSoC CapSense Expansion Board Kit to port E of the development kit.
  - □ Connect the LCD to P8.
- If MiniProg3 is used, disconnect it from the programming port and PC. The MiniProg3 is used for I2C communication. Connect the MiniProg3 to the J3 header on the PSoC CapSense Expansion Board Kit. Make sure that I2C pins SDA, SCL, and GND on the MiniProg3 are mapped to the corresponding I2C pins on the kit; see Figure 3-6 on page 10 for the connections. Note that when MiniProg3 is used for both I2C and programming, the MiniProg3 should be disconnected from the PC every time a connection is changed.

![](_page_23_Picture_1.jpeg)

## 4.4.3 Verify Output

Build and program the code example. If MiniProg3 is used, disconnect the MiniProg3 from the programming header and connect it to I2C connector (J3) to use it for communication with the GUI. Reset the device. See Tuner GUI Use on page 16 and follow the steps.

Monitor the parameter settings and waveforms for different CapSense sensors. Touch any button and observe the corresponding LED turn on. Place a finger on the slider; the nearest LED turns on and the finger position is displayed on the LCD.

Figure 4-6. LCD Showing Slider Position

![](_page_23_Picture_6.jpeg)

![](_page_24_Picture_1.jpeg)

Figure 4-7 shows the monitoring of RawCounts, Baseline, and Difference count (signal) for the Column 0 sensor.

![](_page_24_Figure_3.jpeg)

![](_page_24_Figure_4.jpeg)

**Note** The CY8CKIT-001 PSoC Development Kit gives lesser SNR performance. The kit is not designed to give optimal CapSense performance. Only 1.5 mm overlay should be used with the PSoC Development Kit.

Code Examples

![](_page_25_Picture_1.jpeg)

# A. Appendix

![](_page_26_Picture_1.jpeg)

## A.1 Schematic

![](_page_26_Figure_3.jpeg)

From Expansion Connector of DVK

![](_page_26_Figure_5.jpeg)

To CapSense Connector

![](_page_26_Figure_7.jpeg)

![](_page_27_Picture_0.jpeg)

## A.2 Board Layout

A.2.1 PDC-09801 Top

![](_page_27_Picture_3.jpeg)

A.2.2 PDC-09801 Bottom

![](_page_27_Picture_5.jpeg)

![](_page_28_Picture_0.jpeg)

## A.3 BOM

Item	Qty.	Reference	Value	Description	Manufacturer	Mfr Part Number
				РСВ	Cypress	PDC-09801 Rev**
1	1	J1	40 Pin Header	CONN HEADER .100 DUAL R/A 40POS	Sullins Electronics Corp.	PBC20DBAN
2	1	J2	44 Pin Header	CONN FMALE 44POS DL .1" R/A GOLD	Sullins Electronics Corp.	PPPC222LJBN-RC
3	1	J3	5 Pin Header	CONN HEADER VERT 5POS .100 TIN	Molex/Waldom Electronics	22-28-4050
4	2	R1, R2	Zero	RES 0.0 OHM 1/10W 5% 0805 SMD	Panasonic-ECG	ERJ-6GEY0R00V
Insta	ll on th	ne bttom of th	ne PCB as close	to the corners as possible		
5	2	N/A	BUMPER	BUMPER CLEAR .500X.23" SQUARE	Richco Plastic Co	RBS-3R

## A.4 Board Files

The board files for the CY3280 Matrix Button Module and CY3280 Linear Slider Module are available at C:\Program Files\Cypress\PSoC CapSense EBK\2.0\Hardware.

![](_page_29_Picture_0.jpeg)