



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China





CY8CKIT-031

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

Doc. # 001-66474 Rev. \*F

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

**Copyrights**

© Cypress Semiconductor Corporation, 2011-2013. 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 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.

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



<b>Safety Information</b>	<b>5</b>
<b>1. Introduction</b>	<b>7</b>
1.1 Kit Contents .....	7
1.2 PSoC Creator .....	7
1.3 Getting Started.....	8
1.4 Additional Resources.....	8
1.4.1 Beginner Resources.....	8
1.4.2 Engineers Looking for More .....	8
1.4.3 Learning from Peers.....	8
1.4.4 More Code Examples.....	8
1.5 Documentation Conventions.....	10
<b>2. Installation</b>	<b>11</b>
2.1 CD Installation .....	11
2.2 Hardware .....	12
2.3 Software.....	12
<b>3. Kit Operation</b>	<b>13</b>
<b>4. Code Examples</b>	<b>17</b>
4.1 Migrating the projects to use with CY8CKIT-050 and CY8CKIT-010 .....	17
4.2 Code Example 1: BMM_USB .....	17
4.2.1 Project Description .....	17
4.2.2 Hardware Connections.....	19
4.2.3 Verify Output .....	19
4.3 Code Example 2: SLM_USB .....	20
4.3.1 Project Description .....	20
4.3.2 Hardware Connections.....	20
4.3.3 Verify Output .....	21
4.4 Code Example 3: BMM_I2C_Tuner .....	22
4.4.1 BMM_I2C_Tuning example (Without Tuning) .....	22
4.4.1.1 Project Description .....	22
4.4.1.2 Hardware Connections .....	22
4.4.1.3 Verify Output.....	22
4.4.2 BMM_I2C_Tuning (With Tuning).....	23
4.4.2.1 Project Description .....	23
4.4.2.2 Hardware Connections .....	23
4.4.2.3 Launching Tuner GUI.....	24
4.4.2.4 Verify Output.....	26
4.5 Code Example 4: SLM_I2C_Tuner .....	29
4.5.1 SLM_I2C_Tuning example (Without Tuning) .....	29

4.5.1.1	Project Description .....	29
4.5.1.2	Hardware Connections .....	29
4.5.1.3	Verify Output.....	29
4.5.2	SLM_I2C_Tuning (With Tuning).....	30
4.5.2.1	Project Description .....	30
4.5.2.2	Hardware Connections .....	30
4.5.2.3	Launching Tuner GUI .....	31
4.5.2.4	Verify Output.....	31
<b>A.</b>	<b>Appendix</b> .....	<b>33</b>
A.1	Schematic.....	33
A.2	Board Layout .....	34
A.2.1	PDC-09801 Top Layer .....	34
A.2.2	PDC-09801 Bottom Layer .....	34
A.3	Bill of Materials (BOM).....	35
A.4	Board Files .....	35
A.5	Shielding Electrode.....	35
A.6	Regulatory Compliance Information .....	35
	<b>Revision History</b> .....	<b>37</b>

# Safety Information



## Regulatory Compliance

The CY8CKIT-031 is intended for use as a development platform for hardware or software in a laboratory environment. The board is an open system design, which does not include a shielded enclosure. This may cause interference to other electrical or electronic devices in close proximity.

In a domestic environment, this product may cause radio interference. In this case, the user may be required to take adequate prevention measures. Also, the board should not be used near any medical equipment or RF devices.

Attaching additional wiring to this product or modifying the product operation from the factory default may affect its performance and cause interference with other apparatus in the immediate vicinity. If such interference is detected, suitable mitigating measures should be taken.

The CY8CKIT-031 as shipped from the factory has been verified to meet with requirements of CE as a Class A product.



The CY8CKIT-031 contains electrostatic discharge (ESD) sensitive devices. Electrostatic charges readily accumulate on the human body and any equipment, and can discharge without detection. Permanent damage may occur on devices subjected to high-energy discharges. Proper ESD precautions are recommended to avoid performance degradation or loss of functionality. Store unused CY8CKIT-031 boards in the protective shipping package.



### End-of-Life/Product Recycling

This end-of life for this kit is five years from the date of manufacture, mentioned on the back of the box. Contact your nearest recycler for information on how to disposition the kit.

## General Safety Instructions

### ESD Protection

ESD can damage boards and associated components. Cypress recommends that you perform procedures only at an ESD workstation. If one is not available, use appropriate ESD protection by wearing an antistatic wrist strap attached to chassis ground (any unpainted metal surface) on your board when handling parts.

### Handling Boards

CY8CKIT-031 boards are sensitive to ESD. Hold the board only by its edges. After removing the board from its box, place it on a grounded, static free surface. Use a conductive foam pad if available. Do not slide board over any surface.

# 1. Introduction



Thank you for your interest in the CY8CKIT-031 PSoC® CapSense® Expansion Board Kit. The PSoC CapSense Expansion Board Kit interfaces either of the CY3280 Universal CapSense Module boards with the CY8CKIT-001 PSoC Development Kit, CY8CKIT-030 PSoC 3 Development Kit, or CY8CKIT-050 PSoC 5LP Development Kit. This kit enables you to develop CapSense solutions with the CY3280 Universal CapSense Module boards. The CapSense component in PSoC Creator™ 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 5LP device families. These devices are based on Cypress' Programmable System-on-Chip™ platform for 8-, 16-, and 32-bit applications. PSoC 3/5LP combines precision analog and digital logic with a high-performance CPU. 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 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 thick overlay and one 3-mm thick overlay
- Quick Start Guide
- Resource CD

## 1.2 PSoC Creator

Cypress' 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, PSoC 4 and PSoC 5LP.



## 1.3 Getting Started

To get started, go to [Kit Operation chapter on page 13](#) 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 17](#) explains code examples provided with the kit. The [Appendix chapter on page 33](#) 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](#)

[AN77759 - Getting Started with PSoC 5LP](#)

[PSoC Creator Training](#)

### 1.4.2 Engineers Looking for More

[AN54460 - PSoC 3, PSoC 4, and PSoC 5LP Interrupts](#)

[AN52705 - PSoC 3 and PSoC 5LP - Getting Started with DMA](#)

[AN52701 - PSoC 3 and PSoC 5LP - Getting Started with Controller Area Network \(CAN\)](#)

[AN54439 - PSoC 3 and PSoC 5LP External Crystal Oscillators](#)

[AN52927 - PSoC 3 and PSoC 5LP - 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, PSoC 4 and PSoC 5LP.

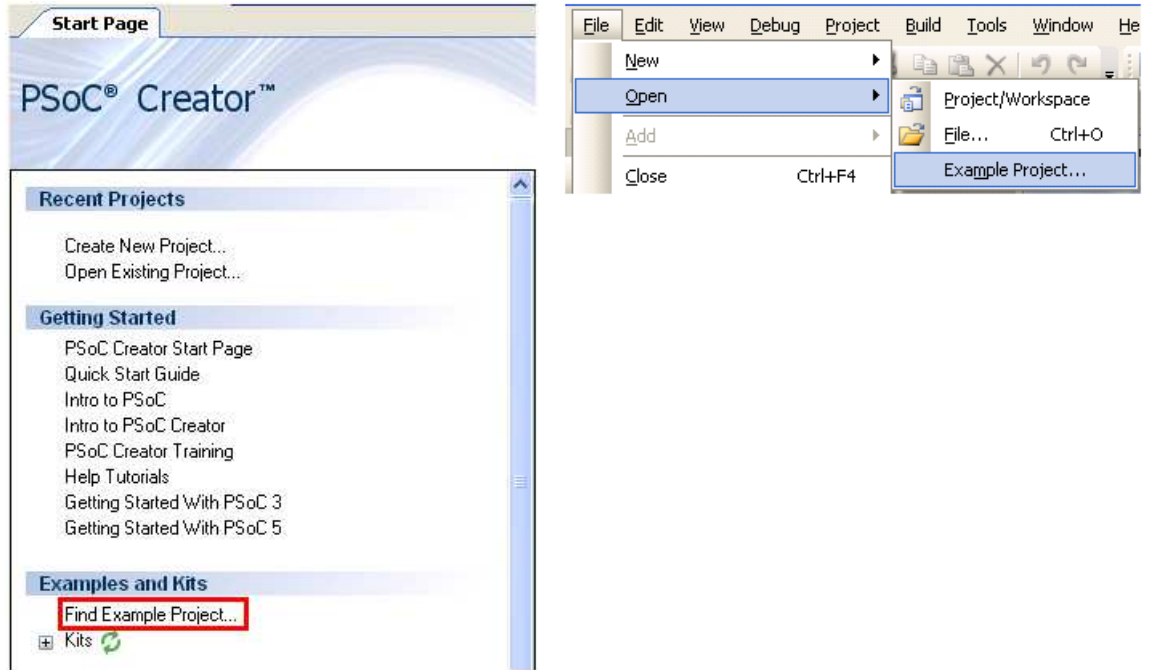
### 1.4.3 Learning from Peers

[Cypress Developer Community Forums](#)

### 1.4.4 More Code Examples

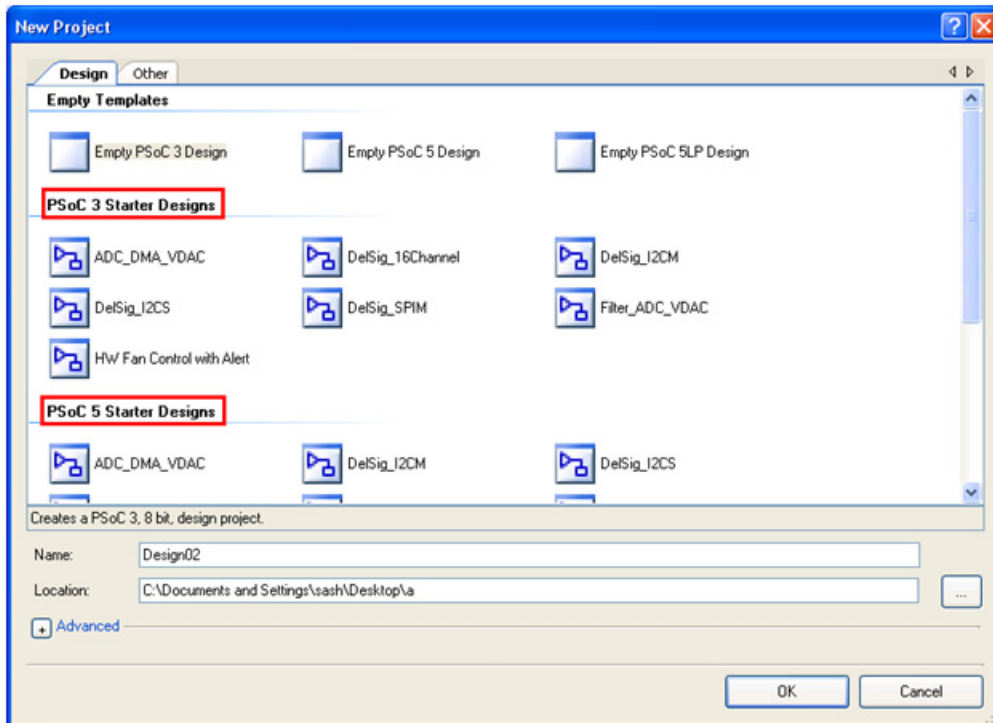
PSoC Creator provides several example projects that make code development fast and easy. To access these example projects, click on **Find Example Project...** under the **Example and Kits** section in the **Start Page** of PSoC Creator or navigate to **File > Open > Example Project...**

Figure 1-1. Find Example Project



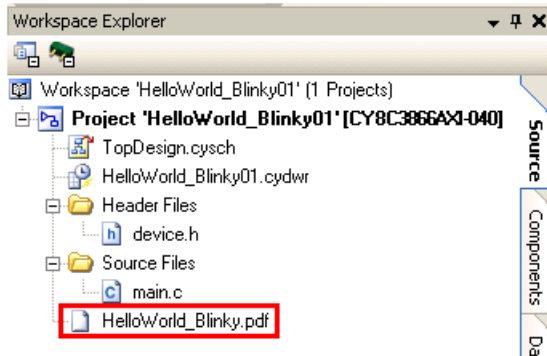
The Find Example Project section has various filters that help you locate the most relevant project. PSoC Creator provides several starter designs. These designs highlight features that are unique to PSoC devices. They allow you to create a design with various components, instead of creating a new empty design; code is also provided. To use a starter design for your project, navigate to **File > New > Project** and select the design required.

Figure 1-2. New Project



The starter designs and the example project contain a PDF within the project that explains the features of the project and its configuration.

Figure 1-3. Project PDF Location



**Note** The example projects and starter designs are designed for CY8CKIT-001 PSoC Development Kit. However, these projects can be converted for use with CY8CKIT-030 PSoC 3 Development Kit or CY8CKIT-050 PSoC 5LP Development Kit by following the procedure in the knowledge base article [Migrating CY8CKIT-001 DVK project to CY8CKIT 030/ 050](#).

Apart from the example projects and starter designs that are available within PSoC Creator, Cypress continuously strives to provide the best support. Click [here](#) to view a growing list of application notes for PSoC 3 and PSoC 5LP.

## 1.5 Documentation Conventions

Table 1-1. Document Conventions for Guides

Convention	Usage
Courier New	Displays file locations, user entered text, and source code: C:\...cd\icc\
<i>Italics</i>	Displays file names and reference documentation: Read about the <i>sourcefile.hex</i> file in the <i>PSoC Designer User Guide</i> .
<b>[Bracketed, Bold]</b>	Displays keyboard commands in procedures: <b>[Enter]</b> or <b>[Ctrl] [C]</b>
File > Open	Represents menu paths: File > Open > New Project
<b>Bold</b>	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.

## 2. Installation



### 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.

Figure 2-1. Kit Menu



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

#### Files Currently on the CD



After the installation is complete, the kit contents are available in the following location :

<Install\_Directory>:\PSoC CapSense EBK\<version>\

The setup installs the following software:

- PSoC Creator
- PSoC Programmer
- Kit Documentation
  - Quick Start Guide
  - Kit Guide
  - Release Notes
- Firmware
  - Code Examples
- Hardware files
  - Schematic
  - Layout
  - BOM

**Note** The hardware files for CY8CKIT-031, CY3280-Matrix Button Module, and CY3280-Linear Slider Module kits are installed in <Install\_Directory>:\PSoC CapSense EBK\<>version>\Hardware.

## 2.2 Hardware

**WARNING** Static discharges from the human body can easily reach very high voltages in the order of kilo volts. This can damage the PSoC 3 or PSoC 5LP device on the development kit. Ensure that any static is discharged before touching the hardware. Follow the below steps to avoid any undesired behavior on the board:

- 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, and Keil compiler, are installed in your computer. If these applications are not installed, the installer installs them in your PC before installing the kit. If Acrobat Reader application is not installed in your PC, then the installer provides the link to install the same and this does not prevent kit installation. Note that Adobe reader is required to view the kit documents.

# 3. Kit Operation



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

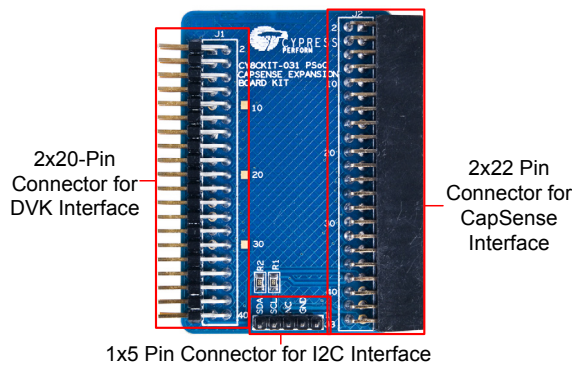


Figure 3-2. CapSense Matrix Button Module

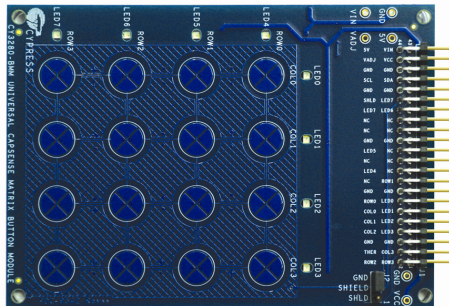
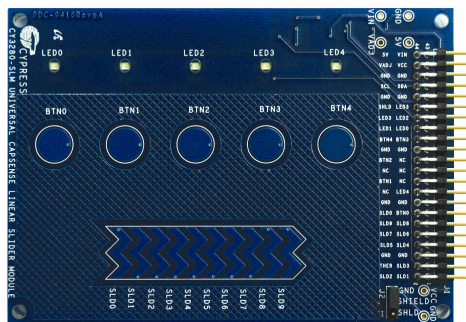


Figure 3-3. CapSense Linear Slider Module



The PSoC CapSense Expansion Board Kit connects to the development kit using a 2x20-pin connector. It connects to the CY3280 Universal CapSense Module Boards using 2x22-pin connector.

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

**Note:** The figure shows the Capsense Expansion board connections when PSoC 3 processor module is used. The same connections need to be done when PSoC 5LP Processor module is used

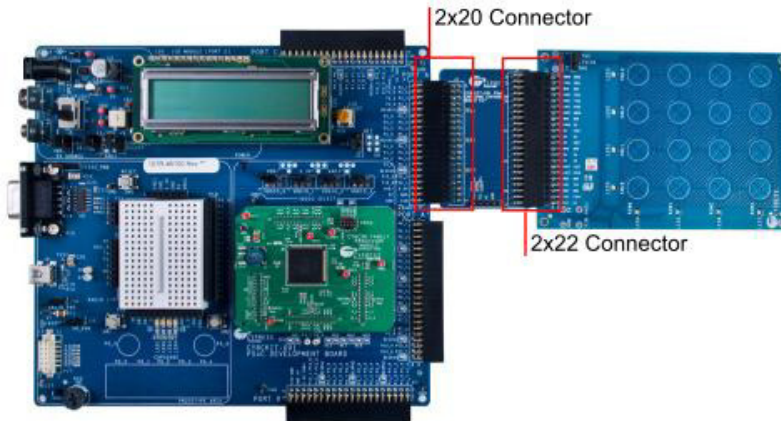
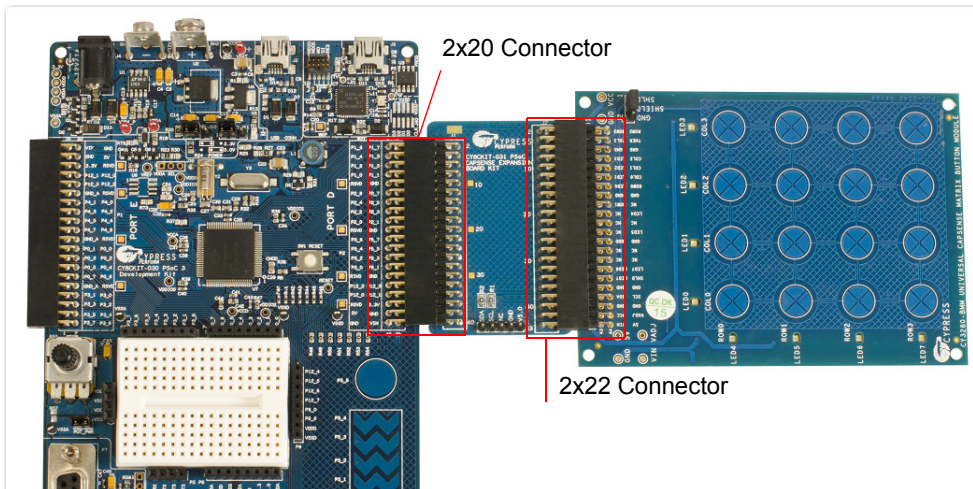


Figure 3-5. PSoC CapSense Expansion Board Kit Interface with PSoC 3 Development Kit and Universal CapSense Matrix Button Module Kit

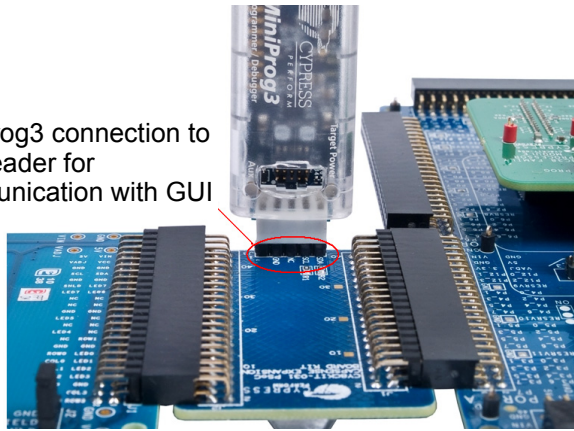
**Note:** The figure shows the Capsense Expansion board connections when PSoC 3 Development Kit is used. The same connections need to be done when PSoC 5LP Development Kit is used.



The 5-pin connector provides the I<sup>2</sup>C interface.

Figure 3-6. I2C Connector

MiniProg3 connection to I2C header for communication with GUI



**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\\_Tuner](#) (works only with MiniProg3)
- [Code Example 4: SLM\\_I2C\\_Tuner](#) (works only with MiniProg3)

The code examples work with the following development kits:

- CY8CKIT-001 PSoC Development Kit (using PSoC 3 Processor module and PSoC 5LP Processor module)
- CY8CKIT-030 PSoC 3 Development Kit
- CY8CKIT-050LP PSoC 5 Development Kit

The PSoC Development Kit (CY8CKIT-001) examples are named with the suffix `_KIT-001`; the PSoC 3 Development Kit examples are named with the suffix `_KIT-030`. The code examples with suffix `_KIT-001` can be used to work with both CY8CKIT-009 (PSoC 3 processor module) and CY8CKIT-010 (PSoC 5LP processor module). Similarly, code examples with suffix `_KIT-030` can be used to work with both CY8CKIT-030 and CY8CKIT-050 as well. See [Migrating the projects to use with CY8CKIT-050 and CY8CKIT-010](#) for more details. Use the appropriate examples according to development kit being used.

All code examples are available in the directory `<Install_Directory>\PSoC CapSense EBK \<version>\Firmware`.

### 4.1 Migrating the projects to use with CY8CKIT-050 and CY8CKIT-010

By default, the project works with PSoC 3. To use the project for PSoC 5LP, change the device to CY8C5868AXI-LP035 (Project > Device selector). If CY8CKIT-001 project is being updated for use with PSoC 5 processor module then apart from the changing the device, change the pin mapping of CMOD pin from P2\_7 to pin 15\_5.

### 4.2 Code Example 1: BMM\_USB

**Note** The code example requires an USB cable to be connected from the PC to the USB connector on the board.

#### 4.2.1 Project Description

This example shows the CapSense Matrix Button Module (BMM) interfacing with the development kit (CY8CKIT-001, CY8CKIT-030, or CY8CKIT-050LP) through the PSoC CapSense Expansion Board. 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 (i.e., the tuning method is set to Auto (Smart-Sense) in the Capsense component), 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:

Table 4-1. Button - Key Mapping

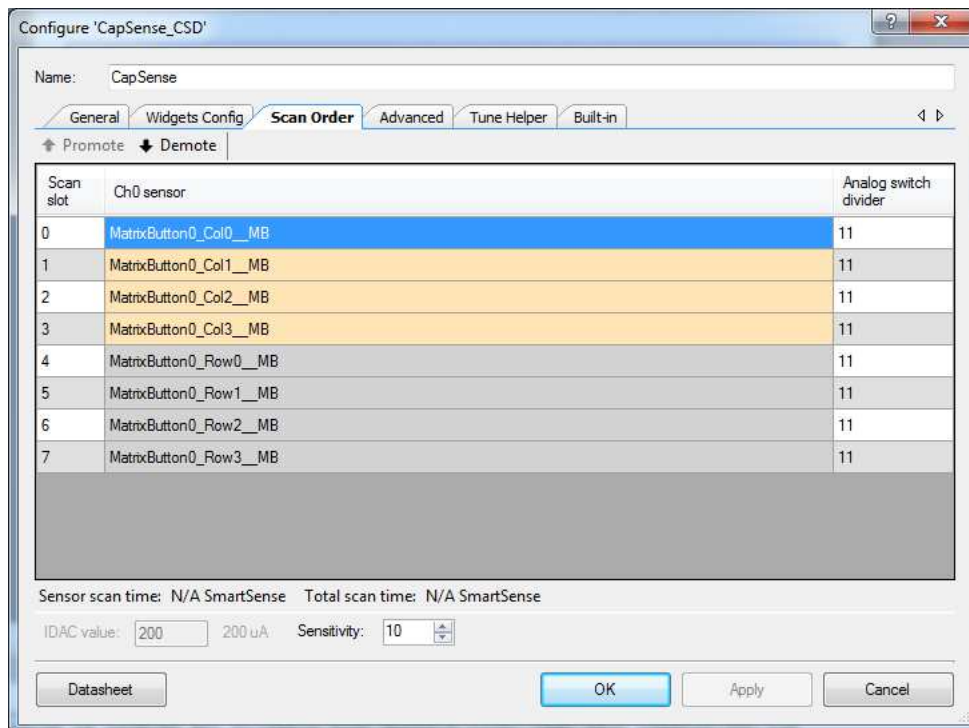
	Col 0	Col 1	Col 2	Col3
Row 0	0	1	2	3
Row 1	4	5	6	7
Row 2	8	9	.	=
Row 3	+	-	*	/

The code waits for the USB cable to be connected to the PC from the USB connector on the board (USB connector J2 on CY8CKIT-030/CY8CKIT-050LP and USB connector J9 on CY8CKIT-001). The code proceeds to CapSense sensor scanning only if USB enumeration is successful. When the USB enumeration is successful, the LED4 on CY8CKIT-030/050 and LED1 on CY8CKIT-001 (a jumper wire connection to P1\_6 is required) on board turns on.

The example is tested to work without any overlay. When an overlay of some thickness is used, the Sensitivity parameter should be changed appropriately for Capsense to work. To configure the Sensitivity parameter, double-click on the **CapSense\_CSD** component. Click on the **Scan Order** tab. Select the individual CapSense sensor and configure sensitivity as required.

The sensitivity parameter indicates the finger capacitance, which depends on the button area, overlay thickness, and dielectric constant of overlay material.

Figure 4-1. Configuring 'CapSense\_CSD'



To use the overlay, remove the adhesive sticker and stick it onto the board in such a way that all the CapSense buttons and sliders are covered. Ensure that no air gaps or bubbles get trapped in between the board and overlay.

#### 4.2.2 Hardware Connections

For the PSoC Development Kit (CY8CKIT-001)

- 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 or PSoC 5LP Development Kit

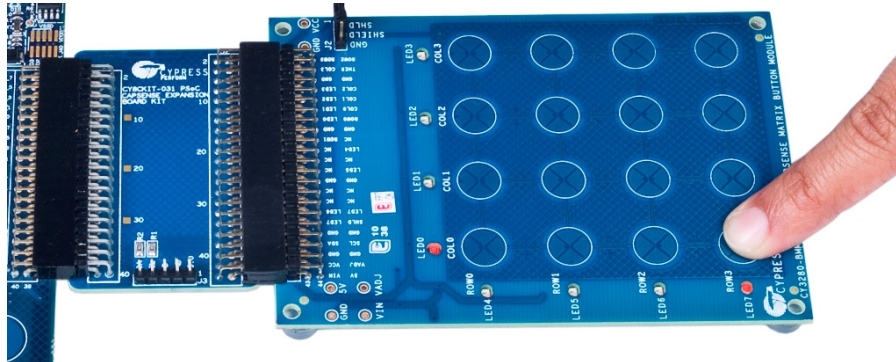
- Connect the CapSense Matrix Button Module to connector J2 of the PSoC CapSense Expansion Board.
- Connect J1 of the PSoC CapSense Expansion Board to port D of the development kit.
- Connect J2 on the CapSense Matrix Button Module to short SHIELD and SHLD.
- Connect jumpers J10 and J11 to position 2 and 3 to use 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 noise on CapSense if MiniProg3 is present.

**Note** See [Shielding Electrode on page 35](#) for more details.

#### 4.2.3 Verify Output

Build and program the code example and reset the device. After device is reset, wait for Enumerate\_LED to turn on, indicating that the system is ready. 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 (see [Table 4-1](#)). Perform different operations on the calculator using CapSense touch buttons.

Figure 4-2. CapSense Matrix Button Module Project



## 4.3 Code Example 2: SLM\_USB

**Note** The code example requires an USB cable to be connected from the PC to the USB connector on the board.

### 4.3.1 Project Description

This example shows the CapSense Linear Slider Module (SLM) interfacing with the development kit (CY8CKIT-001, CY8CKIT-030, or CY8CKIT-050LP) through the PSoC CapSense Expansion Board. The example demonstrates the CapSense slider and button combination. The SLM module has 10 element slider and five buttons. The module also has five LEDs. Touch a button on the module to turn 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 control buttons.

The project uses the auto-tuning feature (i.e., the tuning method is set to Auto (Smart-Sense) in the Capsense component), which sets all CapSense parameters to the 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 - System Volume UP/DOWN

The code waits for the USB cable to be connected to the PC from the USB connector on the board (USB connector J2 on CY8CKIT-030/CY8CKIT-050LP and USB connector J9 on CY8CKIT-001). The code proceeds to CapSense sensor scanning only if the USB enumeration is successful. When the USB enumeration is successful, the LED4 on CY8CKIT-030/050 and LED1 on CY8CKIT-001 (a jumper wire connection to P1\_6 is required) on the board turns on.

The example is tested to work without any overlay. When overlay is used, the sensitivity parameter should be changed appropriately for Capsense to work. To configure the Sensitivity parameter, double-click on the **CapSense\_CSD** component. Click on the **Scan Order** tab. Select the individual CapSense sensor and configure sensitivity as required.

### 4.3.2 Hardware Connections

For the PSoC Development Kit (CY8CKIT-001)

- 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 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 or PSoC 5LP Development Kit

- Connect the CapSense Linear Slider Module to connector J2 of the PSoC CapSense Expansion Board.

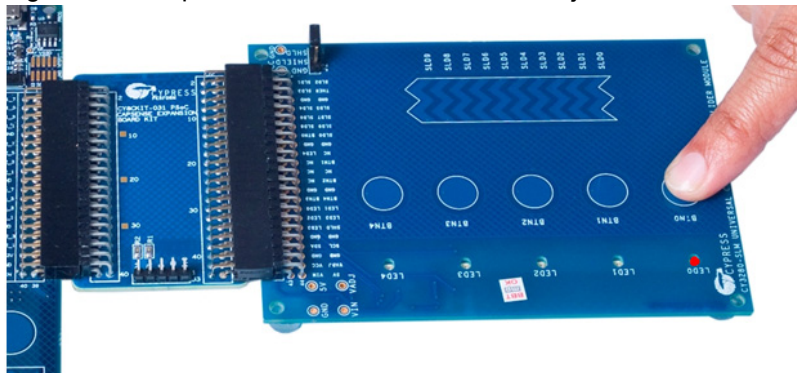
- Connect J1 of the PSoC CapSense Expansion Board Kit to the port D of the development kit.
- Connect J2 on the CapSense Linear Slider Module to short SHIELD and SHLD.
- Connect jumpers J10 and J11 to position 1 and 2 to use the board at 3.3 V.
- Connect the USB cable from J2 on the development kit to 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 noise on CapSense if MiniProg3 is present.

**Note** See [Shielding Electrode on page 35](#) for more details.

### 4.3.3 Verify Output

Build and program the code example and reset the device. After the device is reset, wait for the Enumerate\_LED to turn on, indicating that the system is ready. Touch a button and see the corresponding LED turn on. Place a finger on the slider and see the nearest LED turn on. Open any media player application on the PC. Perform different operations such as Play/Pause, Stop, Mute/Unmute, Next, Previous, and System Volume Control using CapSense Linear Slider Module touch controls.

Figure 4-3. CapSense Linear Slider Module Project



## 4.4 Code Example 3: BMM\_I2C\_Tuner

This code example can be executed in 2 ways: With Capsense Tuning (requires I2C interface using MiniProg3) and Without Capsense Tuning (no additional requirements) i.e., the same project can be used to demonstrate the capsense functionality as well as Capsense tuning using Tuner Helper GUI in PSoC Creator. This is done by commenting/uncommenting the line `#define ENABLE_TUNING` in the `main.c` file of the code example. PSoC creator does not compile the code under `#ifdef` (if defined) statement when `#define` statement is commented (`/*..... */` or `//`). Similarly, when the `#define` statement is removed from comments, the code required for working with Tuner GUI gets compiled. By default, the project is set to work without Capsense tuning by commenting the `#define`.

### 4.4.1 BMM\_I2C\_Tuning example (Without Tuning)

#### 4.4.1.1 Project Description

This code example provides a platform to build CapSense-based projects. This project demonstrates the use of CapSense Matrix Buttons using `Capsense_CSD` component in PSoC Creator.

When a Capsense button is touched, the leds corresponding to the Row and Column gets turned ON. The project uses the auto-tuning feature, which sets all CapSense parameters to the best values automatically. The parameter settings can be monitored in the GUI but cannot be altered because they are set by auto-tuning.

#### 4.4.1.2 Hardware Connections

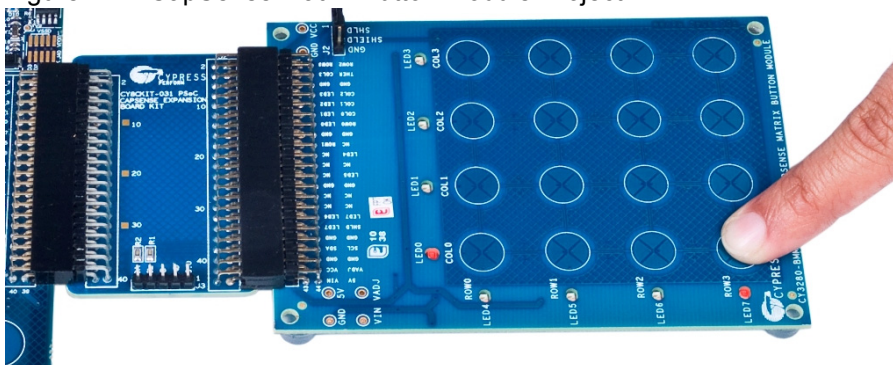
- Connect the CapSense Matrix Button Module to connector J2 of the PSoC CapSense Expansion Board.
- For the PSoC Development Kit, connect J1 of the PSoC CapSense Expansion Board Kit to port A of the development kit.
- For the PSoC 3/5LP Development Kit, connect J1 of the PSoC CapSense Expansion Board 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.
- Build and program the code example and reset the device.

**Note** See Shielding Electrode on page 31 for more details.

#### 4.4.1.3 Verify Output

Touch any button and observe that corresponding row and column LED turns ON.

Figure 4-4. CapSense Matrix Button Module Project



**Note** To evaluate the Capsense tuning part of the code example, a MiniProg3 is required. 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>.

## 4.4.2 BMM\_I2C\_Tuning (With Tuning)

### 4.4.2.1 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 the 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 to '1', as shown in [Figure 4-1 on page 18](#). To configure the Sensitivity parameter, double-click on the **CapSense\_CSD** component. Click on the **Scan Order** tab. Select the individual CapSense sensor and configure the sensitivity as required.

### 4.4.2.2 Hardware Connections

- Connect the CapSense Matrix Button Module to connector J2 of the PSoC CapSense Expansion Board.
- For the PSoC Development Kit (CY8CKIT-001), connect J1 of the PSoC CapSense Expansion Board Kit to port A of the development kit.
- For the PSoC 3 or PSoC 5LP 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.
- Build and program the code example and reset the device.
- 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 15](#) 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.

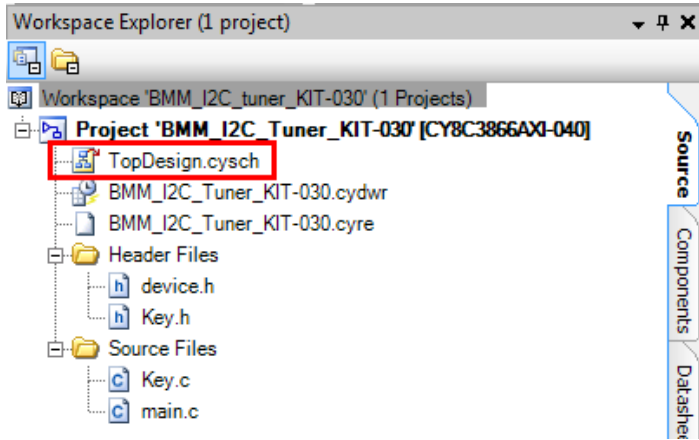
**Note** See [Shielding Electrode on page 35](#) for more details.



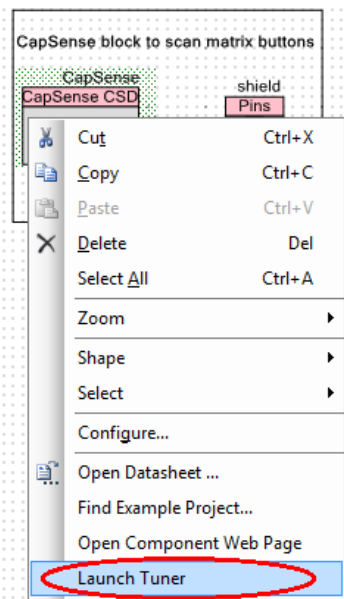
#### 4.4.2.3 Launching Tuner GUI

The Tuner GUI from PSoC Creator should be up and running for the code example to work. To launch the GUI, follow these steps:

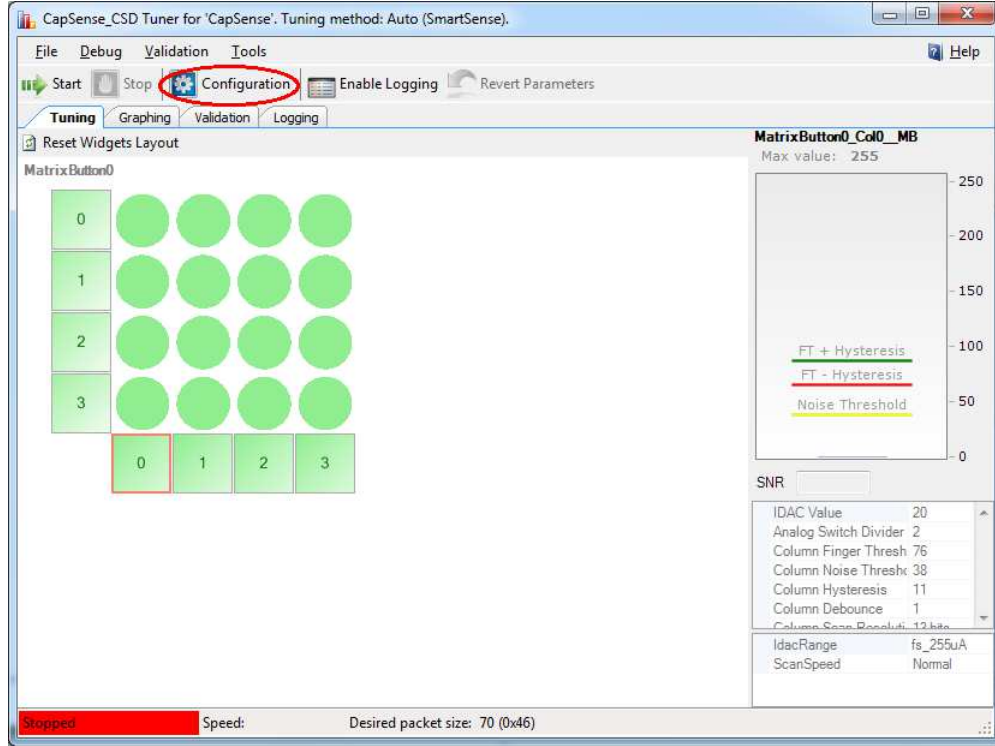
1. Go to the project's *TopDesign.cysch* file.



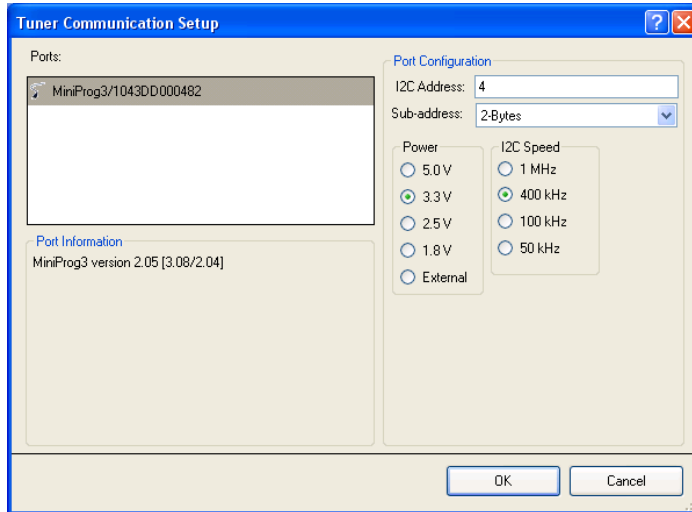
2. To open the tuner, right-click on the **CapSense\_CSD** component in PSoC Creator and click on **Launch Tuner**.



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



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



- 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 and CY8CKIT-050LP.
- Click **OK** to apply the settings.