**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.

# CE210289 - PSoC® 4 CapSense® Linear Slider

## Objective

This example code demonstrates the PSoC 4 CapSense linear slider operation.

## Overview

This code example implements a five-segment CapSense linear slider interface. The linear slider sensor debug data and slider position are read on a PC using the CapSense embedded Tuner GUI via $I^2C$ communication. The CapSense tuner provides a quick and easy method for monitoring and updating CapSense parameters.

## Requirements

**Tool:** PSoC Creator™ 4.1 and later versions

**Programming Language:** C (ARM® GCC 5.4.1, ARM MDK)

**Associated Parts:** All PSoC 4, PSoC 4 BLE and PRoC BLE devices

**Related Hardware:** CY8CKIT-042 and CY8CKIT-042 BLE

## Design

This code example demonstrates the core functionality of the CapSense Component. Figure 1 shows the PSoC Creator schematics of this code example. This example uses the CapSense and EZI2C Slave Components.

The CapSense Component is configured with a five-segment linear slider widget. The project uses the SmartSense (Full Auto-Tune) tuning method to quickly implement a linear slider. The EZI2C Slave is used to monitor the sensor data and slider touch position information on a PC using the CapSense tuner available in the PSoC Creator integrated design environment (IDE) via $I^2C$ communication.

The firmware flowchart is shown in Figure 2.

Figure 1. Top Design

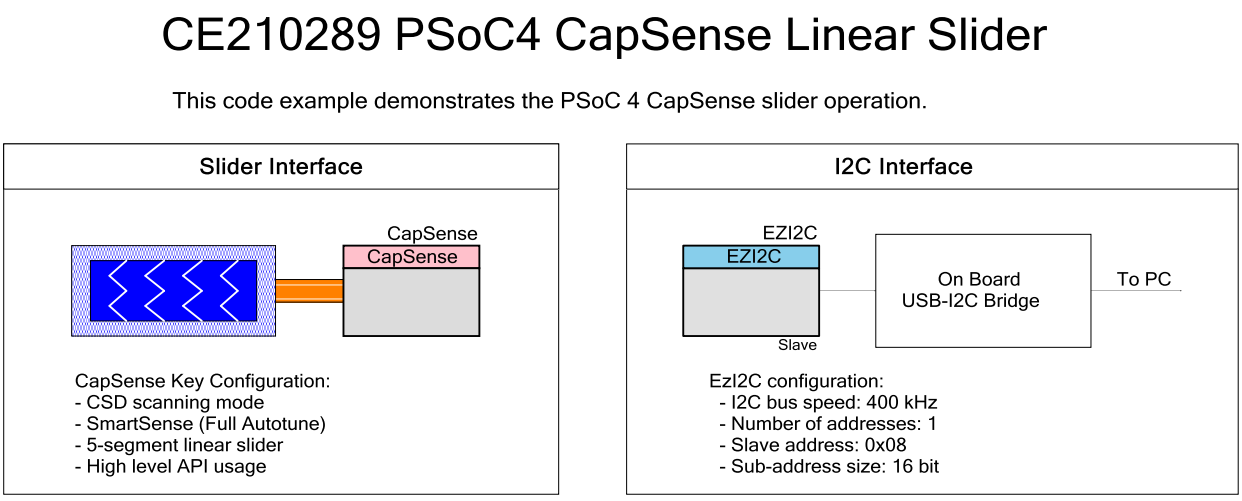
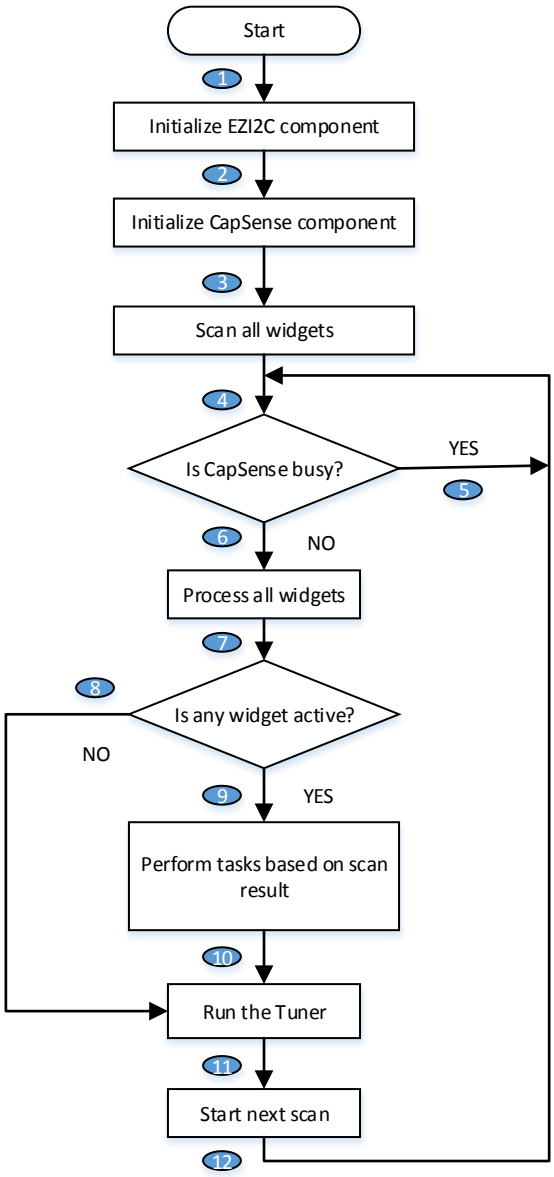


Figure 2. Firmware Flowchart




## Design Considerations

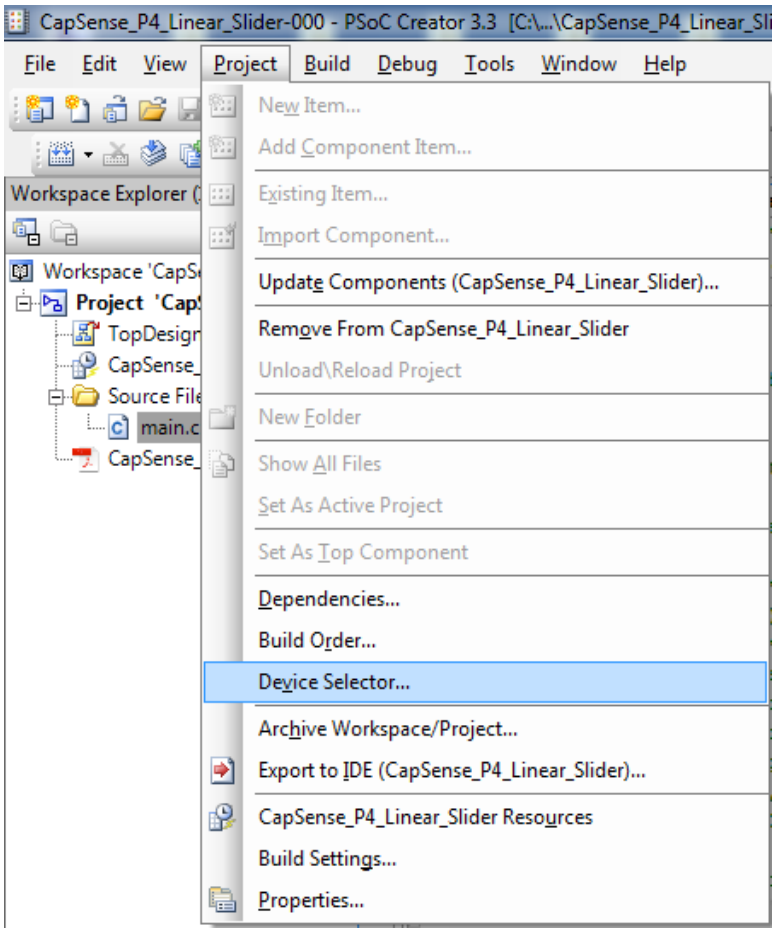
The code example is designed for the PSoC 4200 family and associated CY8CKIT-042 kit. The design is easily portable to other PSoC 4 devices and kits, typically by just changing the device and Components' pin assignment.

To switch from CY8CKIT-042 to other PSoC 4 pioneer kits, follow the steps below:

1. Select the appropriate device with the Device Selector called from the project's context menu (Figure 3). Table 1 lists the device number for each pioneer kit.
   When you select the device, pins are assigned automatically in design wide resource file according to the device selected.

   **Note:** If the assigned Component's pins are not as shown in Table 1 or you want to overwrite the existing pin assignment, double-click the project's design wide resource (DWR) file in the workspace explorer window and assign the pins. See the device datasheet to use other pins.

Figure 3. Select Device Selector from Project's Context Menu




2. Build the project and ensure that there are no errors or warnings.

Table 1. Pin Assignment for the PSoC4 CapSense Linear Slider Project

| Pin name | Development Kit | |
|---|---|---|
| | CY8CKIT-042 (CY8C4245AXI-483) | CY8CKIT-042-BLE (CY8C4247LQI-BL483)/ (CYBL10563-56LQXI) |
| \Capsense: Cmod\ | P4[2] | P4[0] |
| \CapSense: Sns[0]\ (LinearSlider0_Sns0) | P1[1] | P2[1] |
| \CapSense: Sns[1]\ (LinearSlider0_Sns1) | P1[2] | P2[2] |
| \CapSense: Sns[2]\ (LinearSlider0_Sns2) | P1[3] | P2[3] |
| \CapSense: Sns[3]\ (LinearSlider0_Sns3) | P1[4] | P2[4] |

| Pin name | Development Kit | |
|---|---|---|
| | CY8CKIT-042 (CY8C4245AXI-483) | CY8CKIT-042-BLE (CY8C4247LQI-BL483)/ (CYBL10563-56LQXI) |
| \CapSense: Sns[4]\ (LinearSlider0_Sns4) | P1[5] | P2[5] |
| \EZI2C: scl\ | P3[0] | P3[5] |
| \EZI2C: sda\ | P3[1] | P3[4] |

## PSoC Creator Component

Table 2 lists the PSoC Creator Components used in this example, as well as the hardware resources used by each component.

Table 2. List of PSoC Creator Components

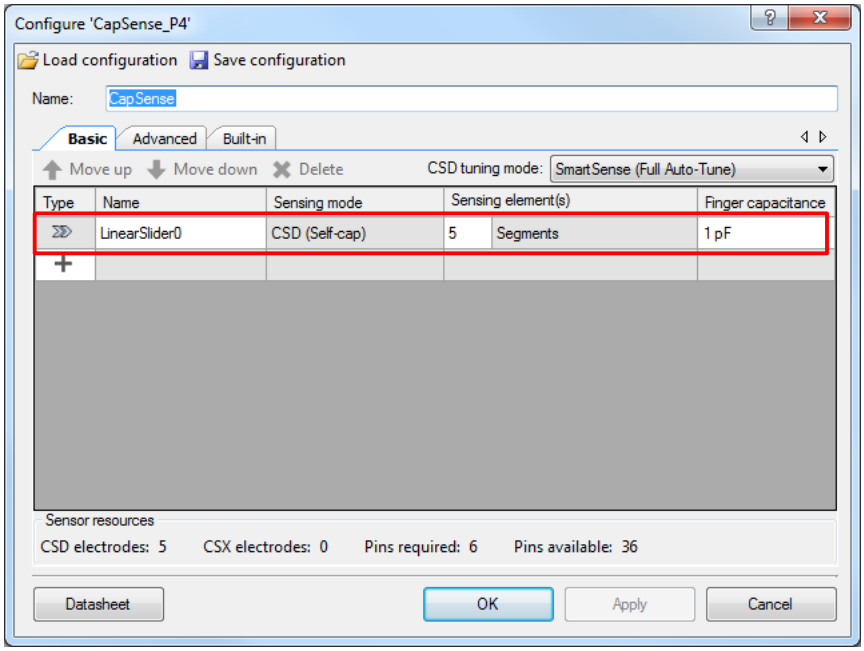
| Component | Instance Name | Component Version | Hardware Resources |
|---|---|---|---|
| CapSense | CapSense | v4.0 | CSD, and 2 GPIO pins |
| EZI2C Save (SCB mode) | EZI2C | v3.20 | SCB, 2 GPIO pins |

### Parameter Settings

#### CapSense

Figure 4 shows the settings for the CapSense Component. See the CapSense Component datasheet for additional information.

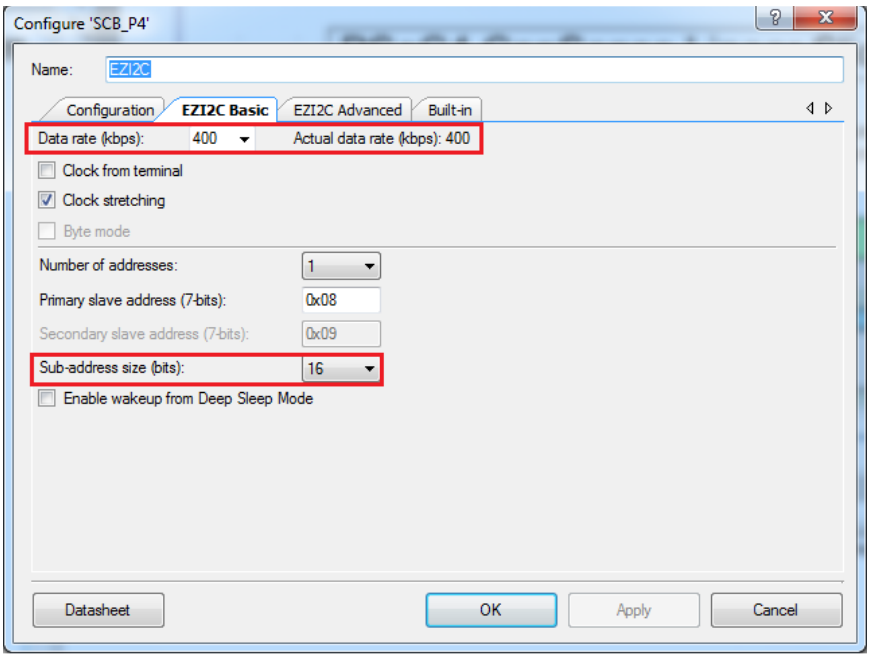
Figure 4. CapSense component's Basic tab




#### EZI2C Component

Figure 5 shows the settings for EZI2C Component. See the SCB Component datasheet for additional information.

Figure 5. EZI2C Slave Component Basic Settings

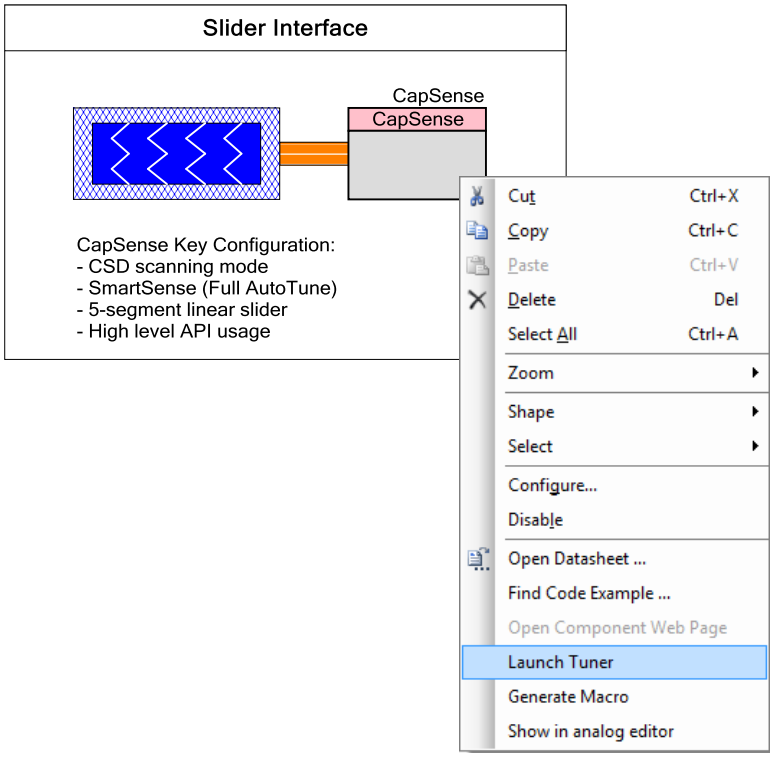



## Operation

After you build and install the example in CY8CKIT-042, test the example by doing the following:

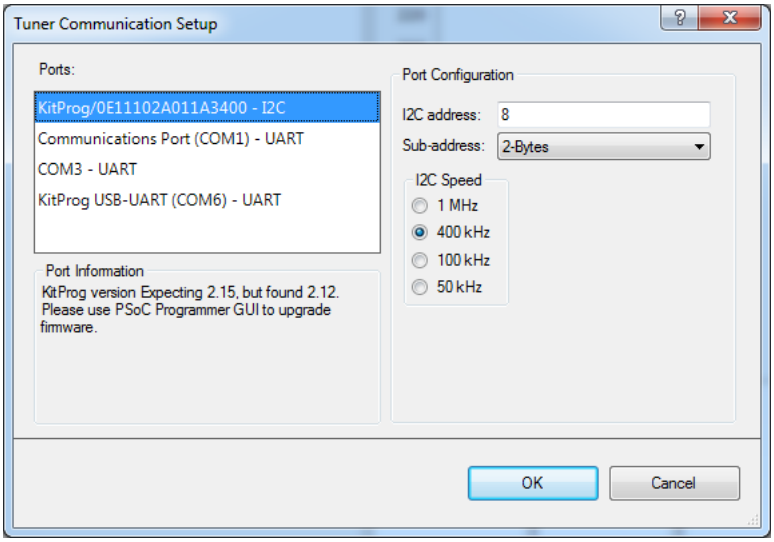
1. Launch the Tuner GUI. Right-click the CapSense Component as shown in (Figure 6). Select **Launch Tuner** in the menu.

Figure 6. Launch Tuner GUI



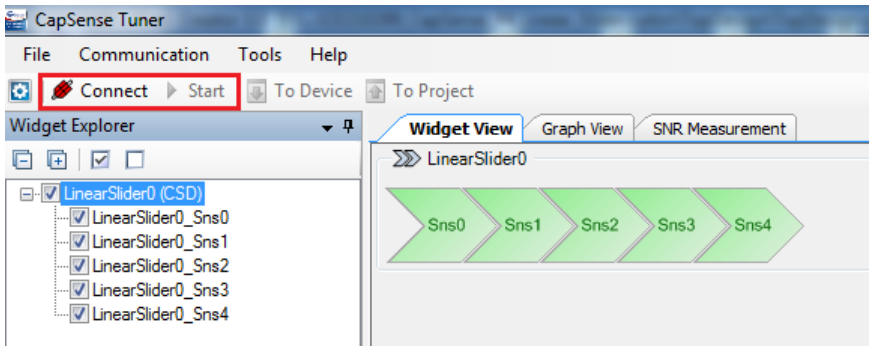

2. Navigate to **Tools/Protocol Settings** in the Tuner GUI menu to set up the $I^2C$ communication (Figure 7).
   - Select the I2C device of the kit
   - Choose the I2C address, sub-address size, and speed as shown in Figure 7.

Figure 7. Setting Up I2C Communication




3. In the tuner GUI, click the Connect button followed by Start button (Figure 8).

Figure 8. Starting Communication




4. After establishing the $I^2C$ communication between the device and Tuner GUI, you can observe the sensor data and the slider position when touching the slider. Figure 9 and Figure 10 show sensor debug data such as difference count, raw count, and baseline.

Figure 9. Tuner: Widget View

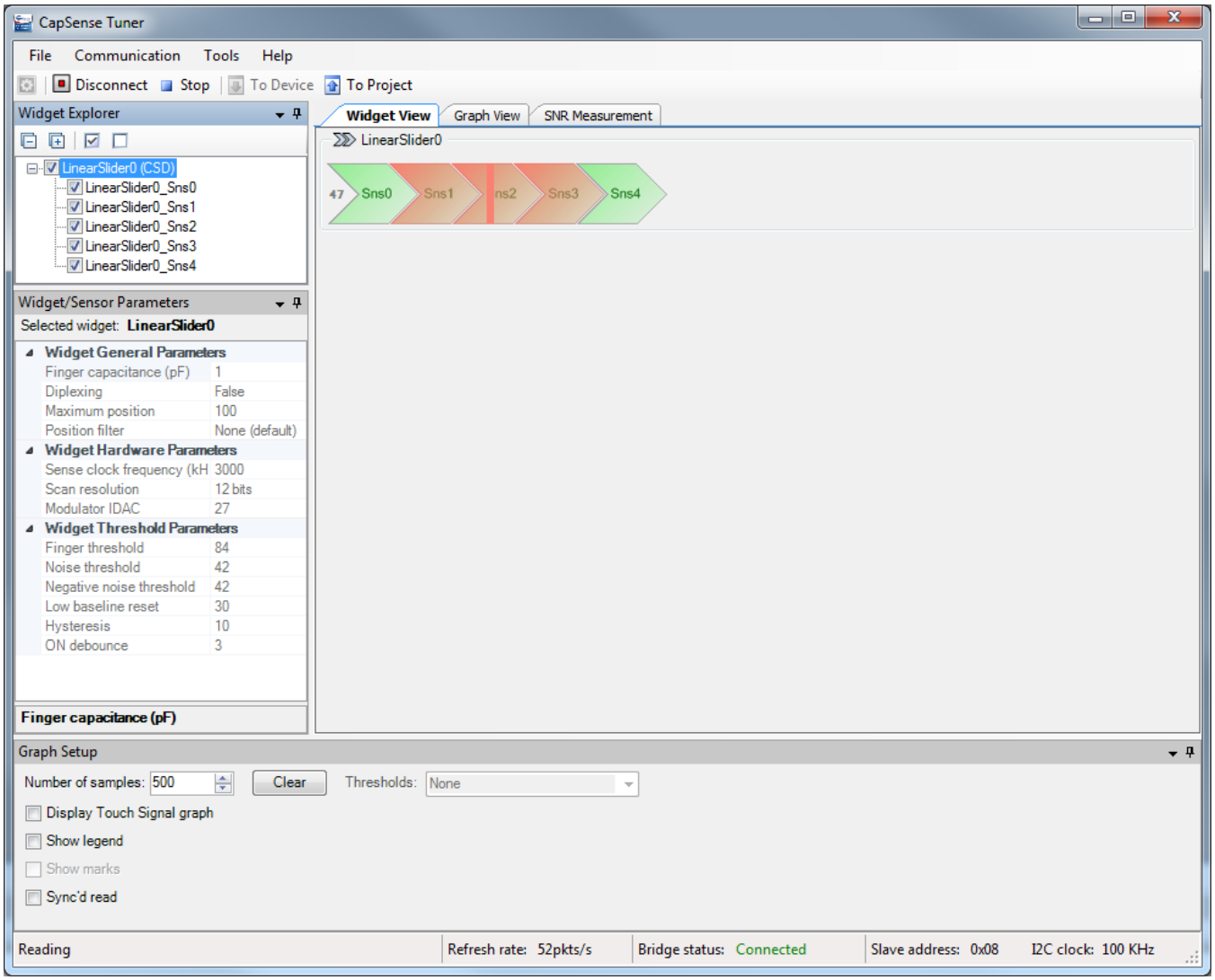
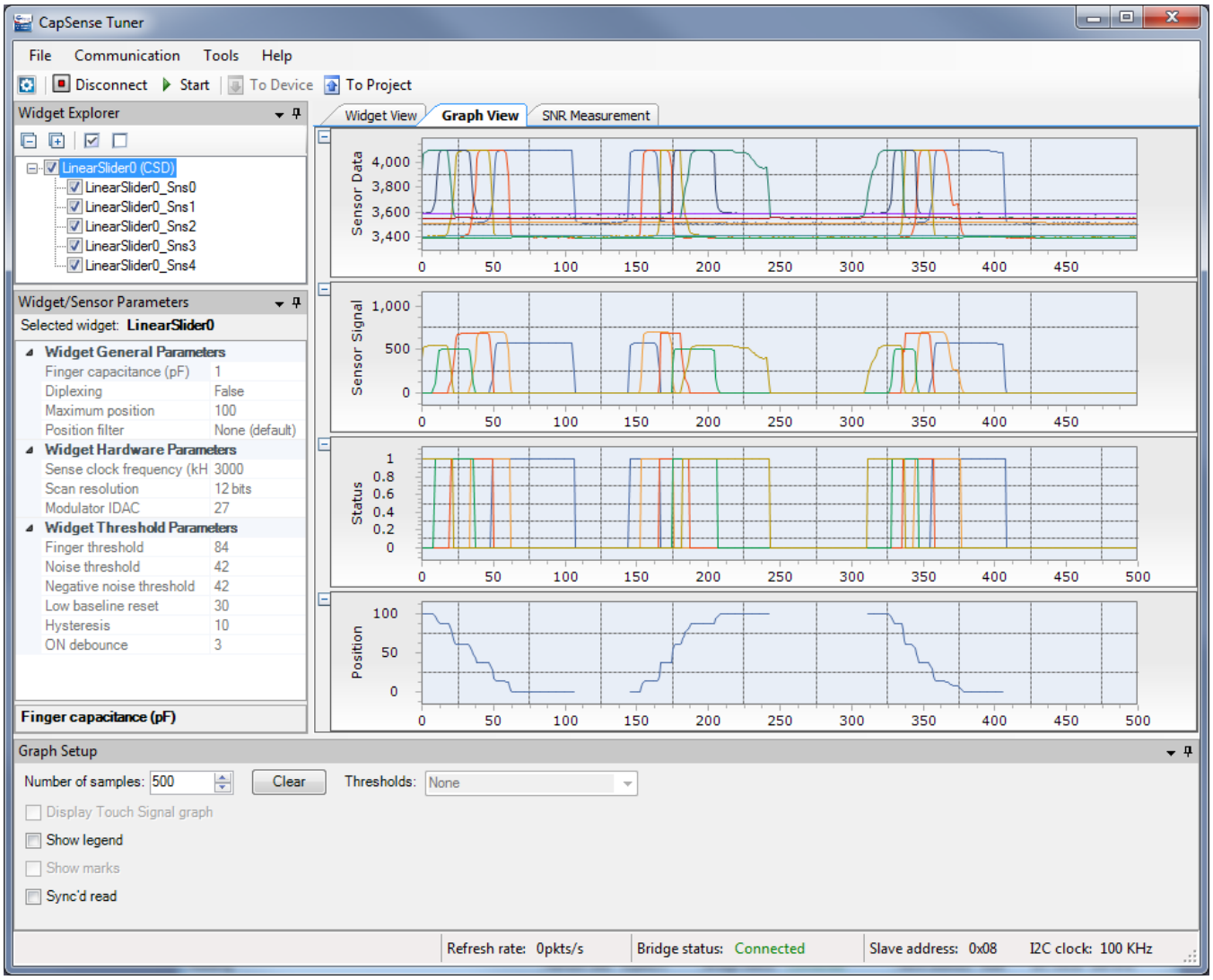


Figure 10. Tuner: Graph View



## Related Documents

Table 3 lists relevant application notes, code examples, device datasheets, and PSoC Creator Component datasheets.

Table 3. Related Documents

| | | |
|---|---|---|
| **Application Notes** | | |
| AN79953 | Getting Started with PSoC 4 | Describes the PSoC 4 and how to build a first PSoC Creator project. |
| AN85951 | AN85951 - PSoC® 4 CapSense® Design Guide | Design Guide shows how to design capacitive touch sensing applications with the PSoC 4 |
| **Code Examples** | | |
| CE210291 | PSoC® 4 CapSense® One Button | |
| CE210290 | PSoC 4 CapSense Low-Power Ganged Sensor | |
| **PSoC Creator Component Datasheets** | | |
| CapSense | Supports capacitive touch sensing | |
| EZI2C Slave | Supports I2C slave operation | |
| **Device Documentation** | | |
| PSoC 4 Datasheets | PSoC 4 Technical Reference Manuals | |
| **Development Kit (DVK) Documentation** | | |
| CY8CKIT-042 PSoC® 4 Pioneer Kit | | |
| CY8CKIT-042-BLE Bluetooth® Low Energy (BLE) Pioneer Kit | | |

## PSoC Resources

Cypress provides a wealth of data at www.cypress.com to help you select the right PSoC device, 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:

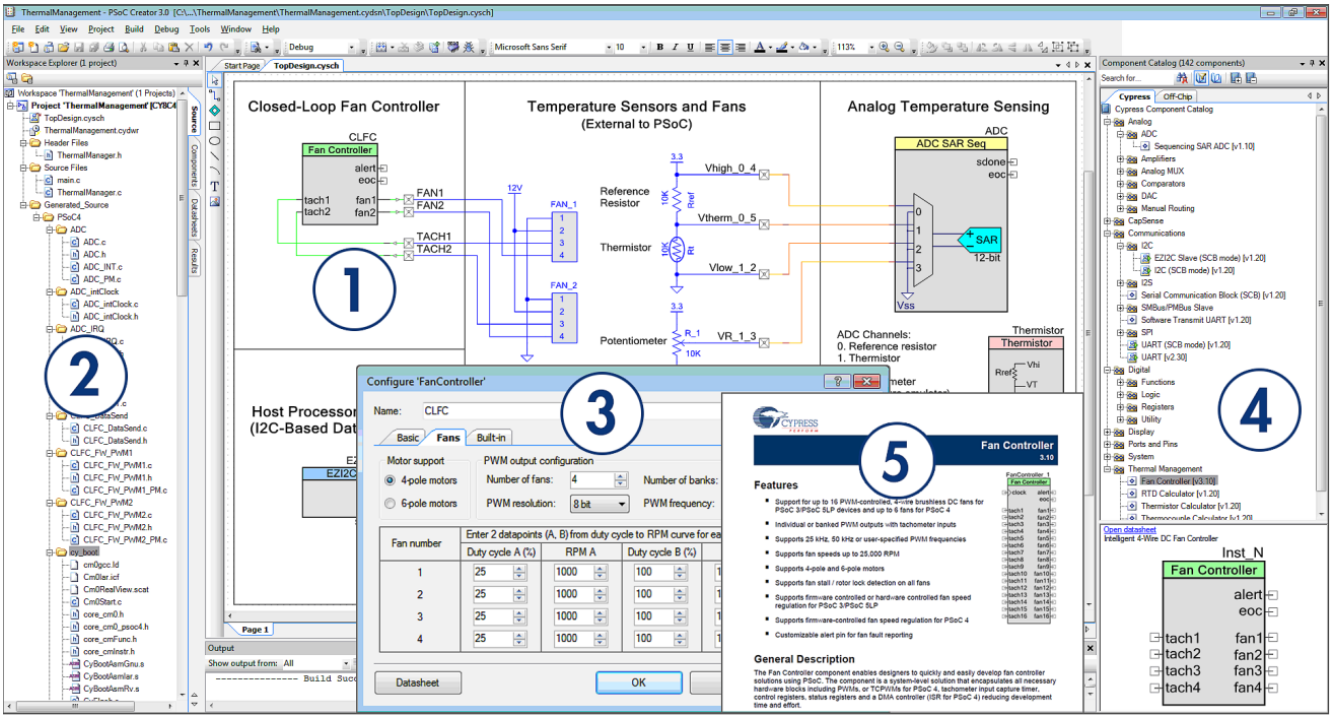
- **Overview:** PSoC Portfolio, PSoC Roadmap
- **Product Selectors:** PSoC 1, PSoC 3, PSoC 4, or PSoC 5LP. In addition, PSoC Creator includes a device selection tool.
- **Datasheets**: Describe and provide electrical specifications for the PSoC 4000, PSoC 4000S, PSoC 4100S, PSoC 4100, and PSoC 4200, PSoC 4xx7 BLE, PSoC 4200-M and PSoC Analog Coprocessor device families
- **CapSense® Design Guide:** Learn how to design capacitive touch-sensing applications with the PSoC 4 family 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. PSoC Creator provides additional code examples.
- **Technical Reference Manuals (TRM):** Provide detailed descriptions of the architecture and registers in each PSoC 4 device family.
- **PSoC Training Videos**: These videos provide step-by-step instructions on getting started building complex designs with PSoC.
- **Development Kits:**
  - CY8CKIT-040, CY8CKIT-041-40xx, CY8CKIT-041-41xx, CY8CKIT-042, CY8CKIT-042-BLE, CY8CKIT-044 and CY8CKIT-048 PSoC Pioneer Kits are easy-to-use and inexpensive development platforms. These kits include connectors for Arduino™ compatible shields and Digilent® Pmod™ daughter cards.
  - CY8CKIT-049 is a very low-cost prototyping platform for sampling PSoC 4 devices.
  - CY8CKIT-001 is a common development platform for all PSoC family 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 11 – 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 11. PSoC Creator Features



# Document History

Document Title: CE210289 - PSoC® 4 CapSense® Linear Slider

Document Number: 002-10289

| Revision | ECN | Orig. of Change | Submission Date | Description of Change |
|---|---|---|---|---|
| ** | 5142891 | AKSM | 02/22/2016 | New code example |
| *A | 5442393 | VMED | 09/20/2016 | Updated components to the latest versions available in PSoC Creator 4.0 |
| *B | 5638792 | VMED | 03/02/2017 | Updated components to the latest versions available in PSoC Creator 4.1<br>Minor edits throughout the document<br>Updated template |
| *C | 5740011 | AESATP12 | 05/26/2017 | Updated logo and copyright. |

## 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

## Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

## Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2016-2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

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

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