CE210311 - CapSense® ADC Sequential

Objective

This code example demonstrates how to use the CapSense_ADC Component to scan CapSense® sensors and measure the input voltage on any pin.

Overview

The CapSense block in PSoC® 4 S-Series supports both capacitive sensing and input voltage measurement. The capacitive sensing mode is used for touch sensing and ADC mode is used for temperature measurement. This code example scans a single button sensor and measures voltage across the potentiometer after the button sensor is scanned.

The CapSense tuner reads button sensor data using I^2C communication. The PC displays the ADC conversion result using UART (SW_Tx_UART) communication. The onboard LED on the kit displays the conversion result. The brightness of the LED varies based on the input voltage level.

Requirements

Tool: PSoC Creator™ 3.3 SP2 or higher

Programming Language: C (GCC 4.9)

Associated Parts: PSoC 4000S and PSoC 4100S

Related Hardware: CY8CKIT-041-40xx and CY8CKIT-041-41xx

Design

The code example consists of CapSense_ADC, SW_Tx_UART, EZI2C, and PWM Components and one digital output pin. Figure 1 shows the Component schematics for this project.

The CapSense Component is configured with a one-button widget. The project uses the CapSense sigma delta (CSD) manual tuning mode with IDAC auto-calibration enabled. The EZI2C Component sends the sensor data to the external host or tuner via I^2C. The design uses the SW_Tx_UART Component to send the ADC data to the PC. The PWM Component changes the LED brightness based on the output of the CapSense_ADC Component. Figure 2 shows the firmware flow chart.

Figure 1. Component Schematics

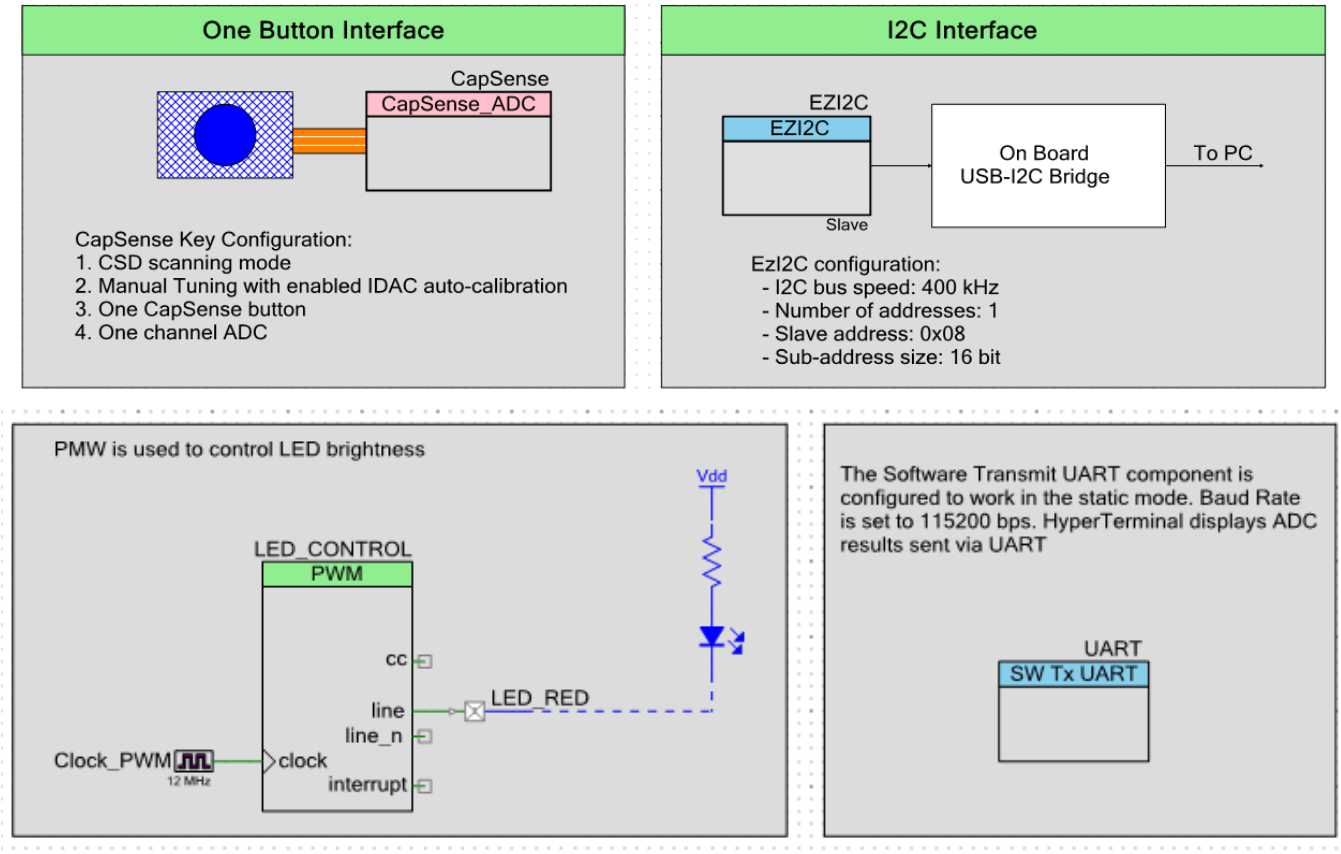
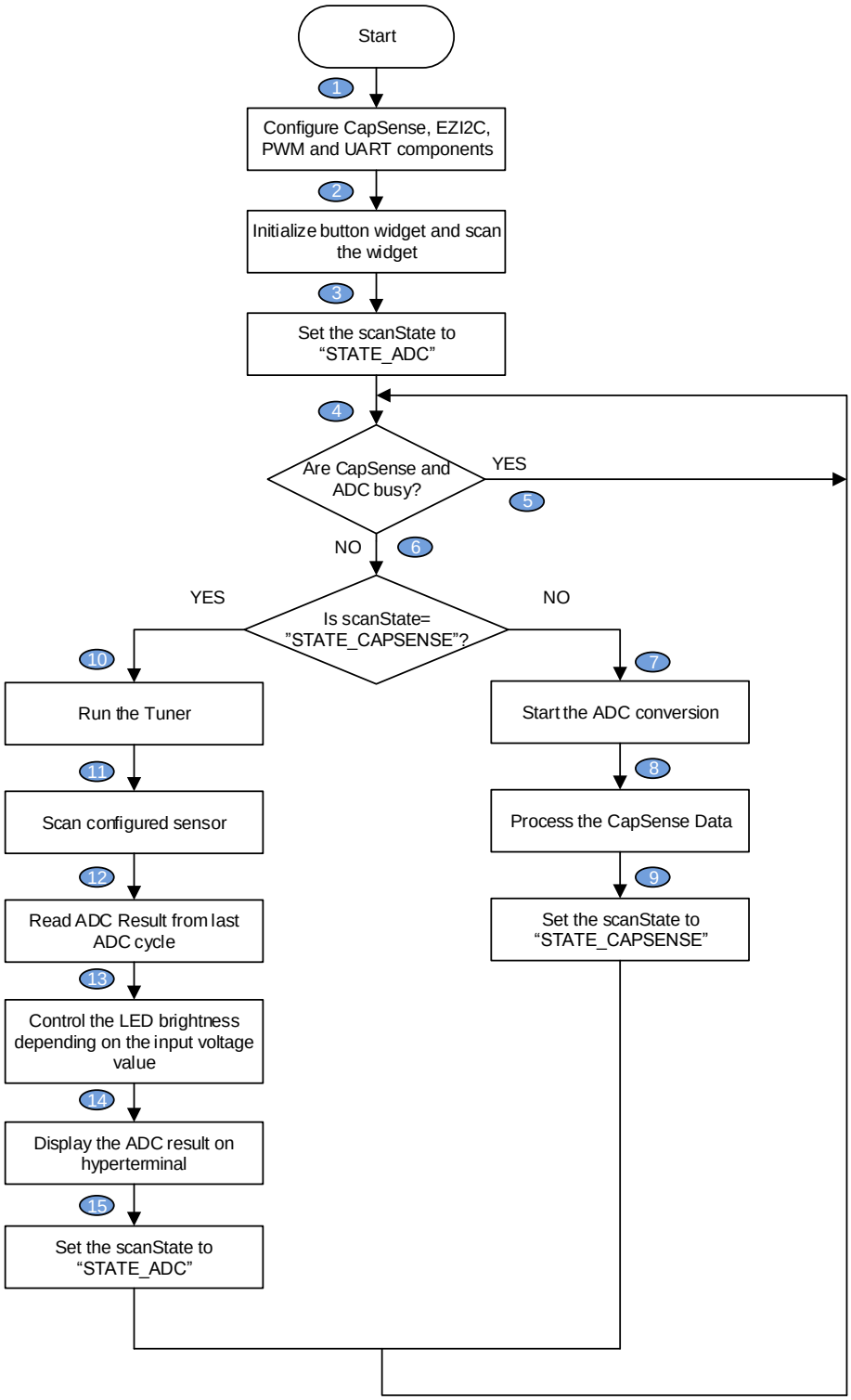


Figure 2. Firmware Flow Chart



Design Considerations

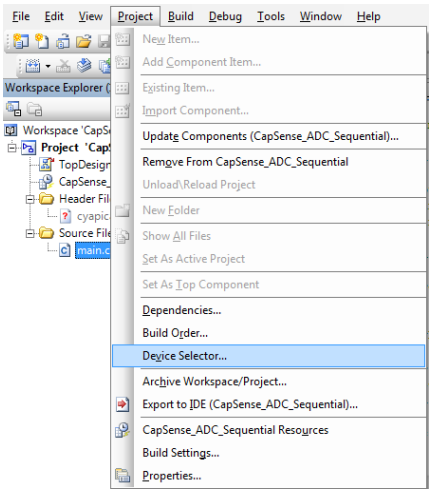
The code example is designed for the PSoC 4000S family and the associated CY8CKIT-041-40xx kit. The design is easily portable to other PSoC 4 devices and kits, typically, by just changing the device and pin assignments.

To switch from the CY8CKIT-041-40xx to other PSoC 4 Pioneer kits, follow these steps:

1. Select the appropriate device with a Device Selector called from the project's context menu (Figure 3). Table 1 lists the device number for each Pioneer kit.
2. When you select the device, PSoC Creator assigns pins automatically for the selected device, in the design wide resource file.

   Note: If the assigned Component 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 file in the Workspace Explorer window and assign the pins. Refer to the device datasheet for information on pin assignments.

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



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

Table 1. Pin Assignment for the CapSense ADC Sequential Project

Pin Name	Development Kit	
	CY8CKIT-041-40xx (CY8C4045AZI-S423)	CY8CKIT-041-41xx (CY8C4146AZI-S433)
\CapSense :AdcInput\	P2[4]	P2[4]
\CapSense :Cmod\ (Cmod)	P4[1]	P4[1]
\CapSense :Sns\ (Button0_Sn0)	P0[1]	P0[1]
\EZI2C :scl\	P3[0]	P3[0]
\EZI2C :sda\	P3[1]	P3[1]
\UART :tx\	P0[5]	P0[5]
LED_RED	P3[4]	P3[4]

PSoC Creator Components

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	Hardware Resources
CapSense	CSD fixed block, interrupts and two pins
EZI2C	Two pins
LED_CONTROL	One TCPWM
UART	One pin
LED_RED	One pin

Parameter Settings

CapSense

Figure 4 and Figure 5 show the settings for the CapSense_ADC Component. This code example requires a CapSense button, which is tuned using manual tuning.

Figure 4. CapSense_ADC Component's Basic Tab

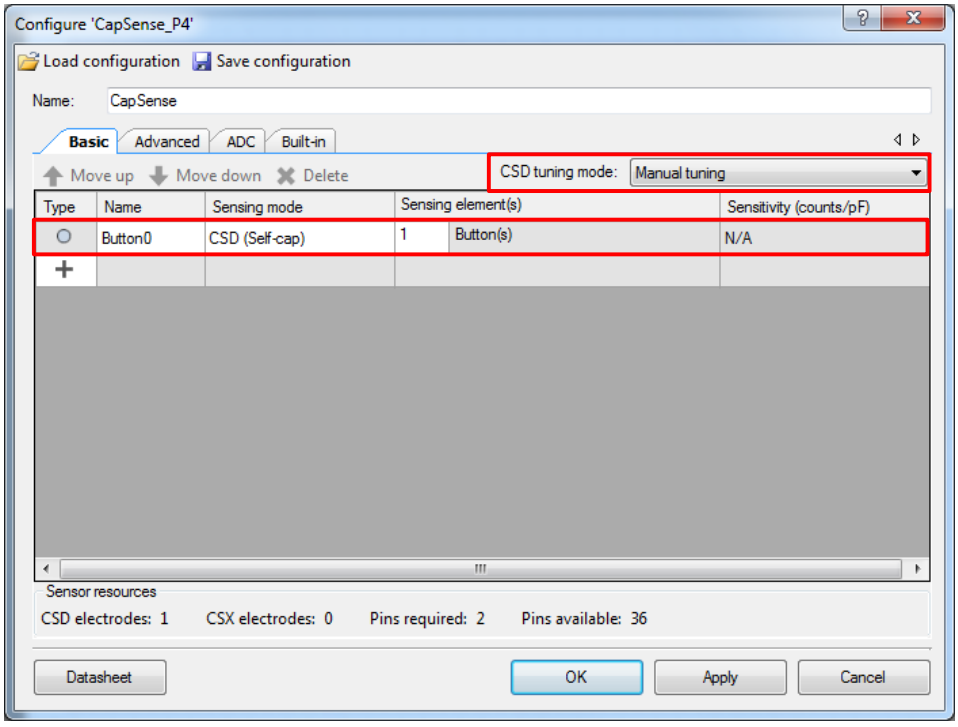
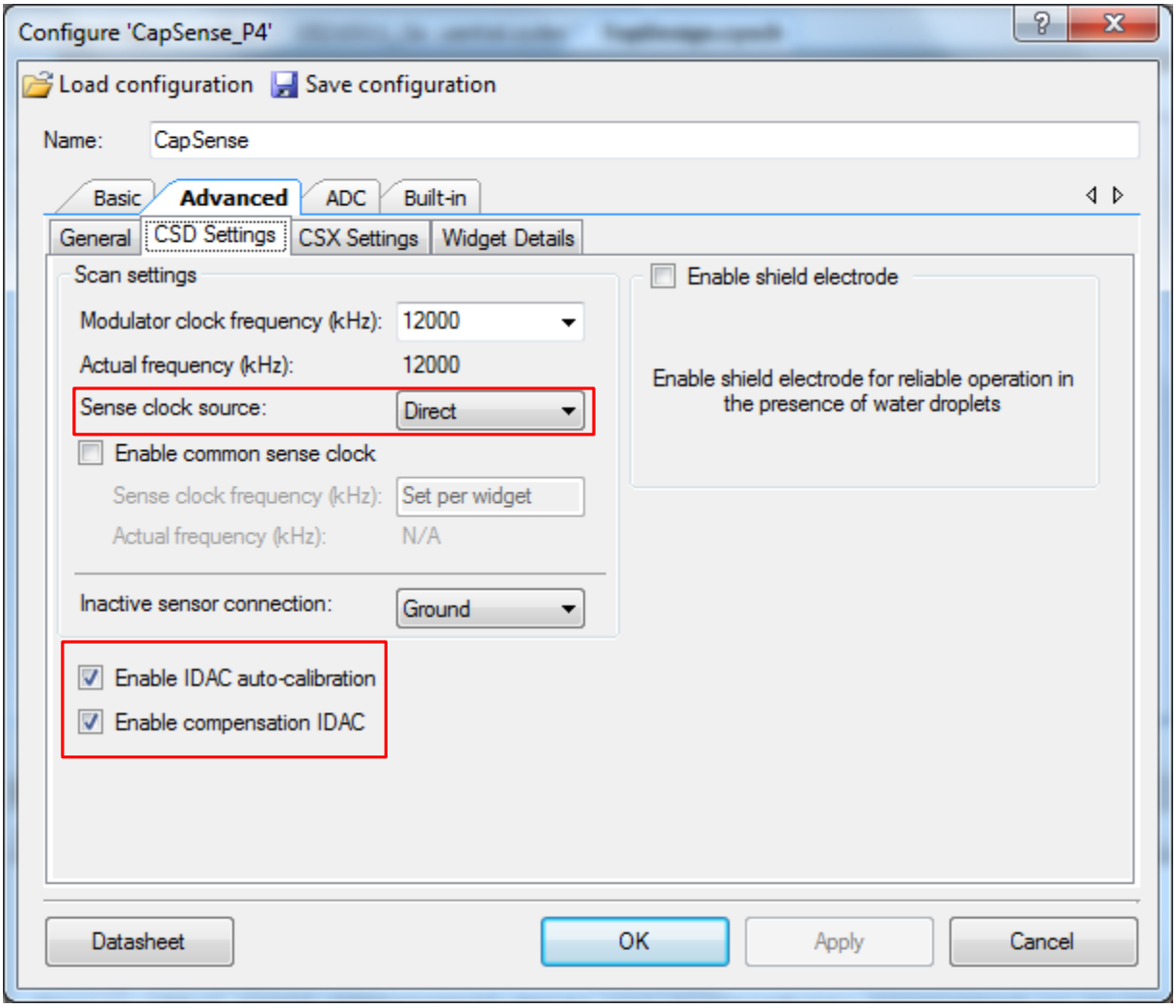


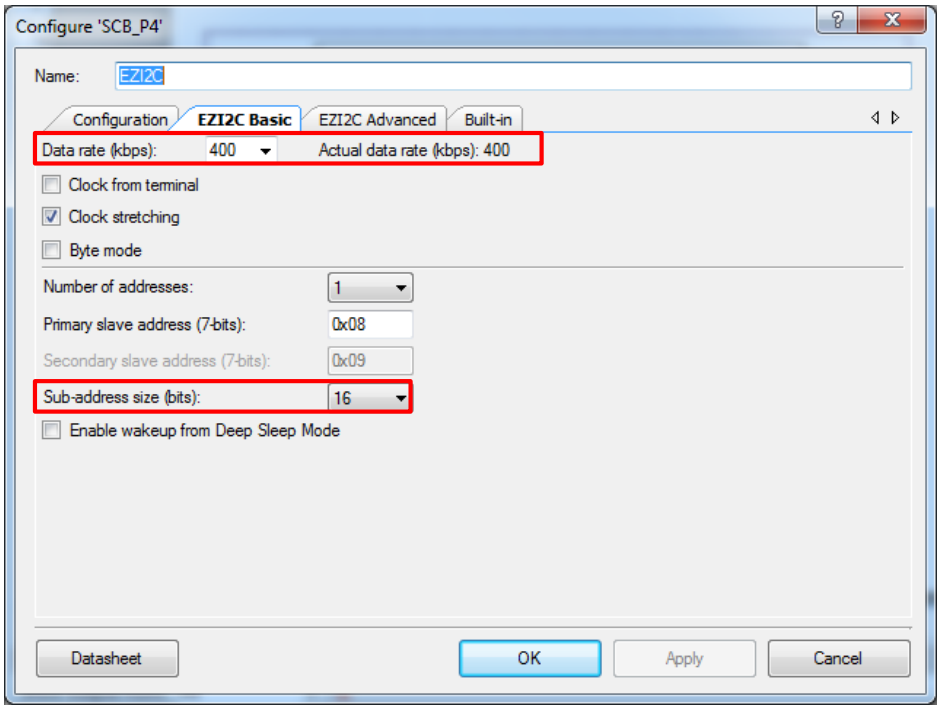
Figure 5. CapSense_ADC Component Customizer Advanced Tab (CSD Settings)



EZI2C Component

Figure 6 shows the settings for the EZI2C Component. Select the 400 kbps data rate; it is optimum for this application. Set sub-address size of 16 bits to allow the master to access data offsets between 0 and 65535.

Figure 6. EZI2C Component's Basic Tab

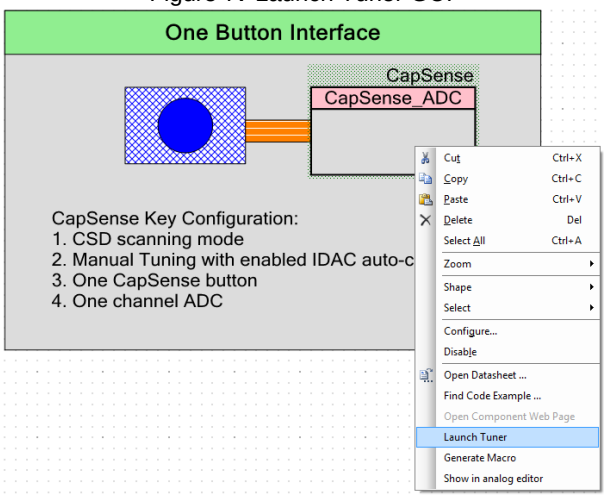


Operation

After you build and install the example in the CY8CKIT-041-40xx kit, test the example by doing the following:

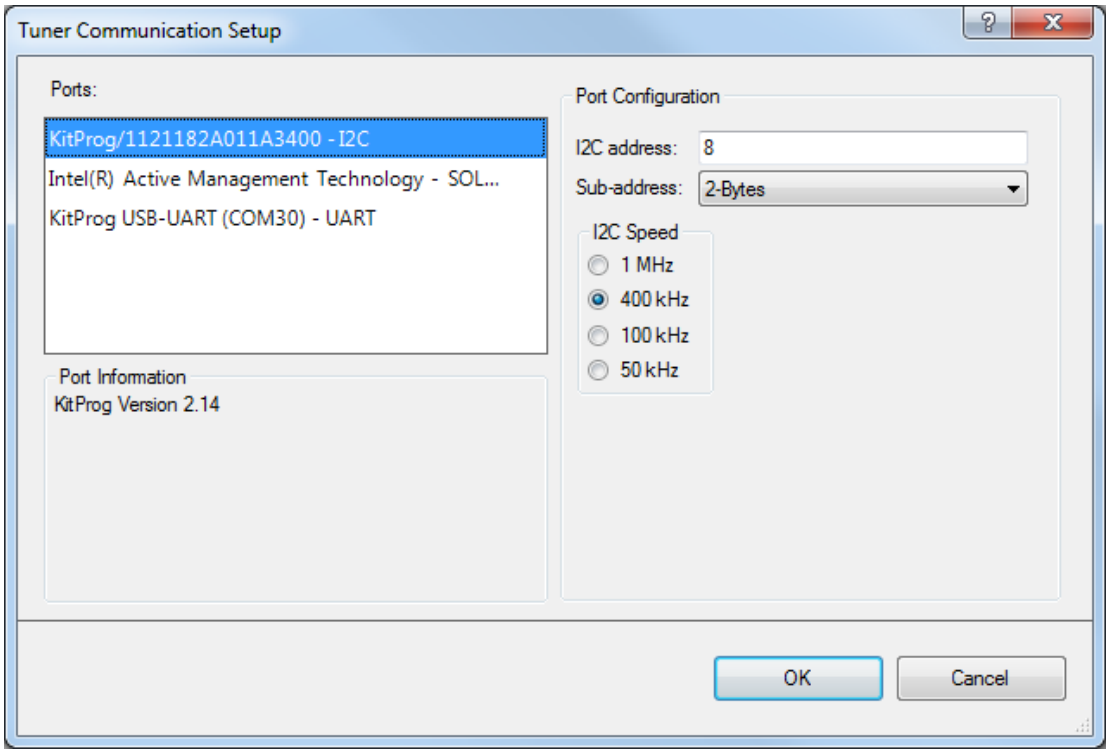
1) Launch the Tuner GUI. Right-click the CapSense Component, as shown in Figure 7. Then, select **Launch Tuner** in the menu.

Figure 7. Launch Tuner GUI



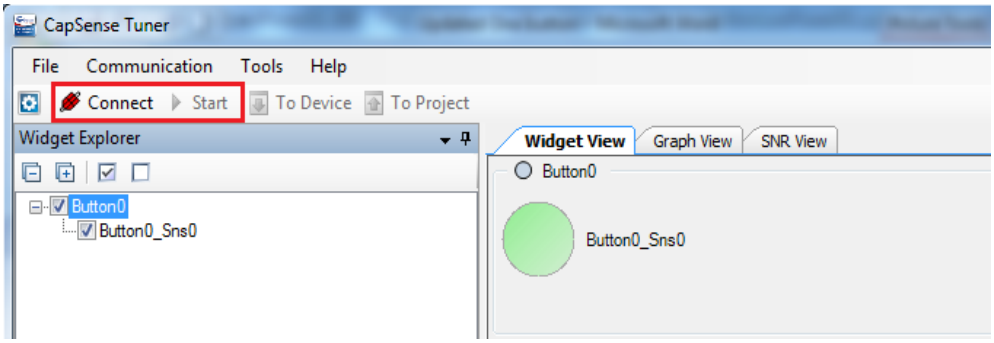
2) Navigate to **Tools** > **Protocol Settings** in the Tuner GUI menu to set up the I²C communication (Figure 8).
   - Select the I²C device of the kit.
   - Choose the I²C address, sub-address size, and speed similar to the EZI2C configuration (Figure 6).

Figure 8. Setting Up I2C Communication



3) Click **Connect**; then, click **Start** (Figure 9).

Figure 9. Starting Communication



4) After establishing I²C communication between the device and Tuner GUI, touch the slider to observe the sensor data and touch position.

5) The Tuner GUI displays the scanning results for all the sensing elements. See Figure 10 and Figure 11.

Figure 10. Tuner: Widget View

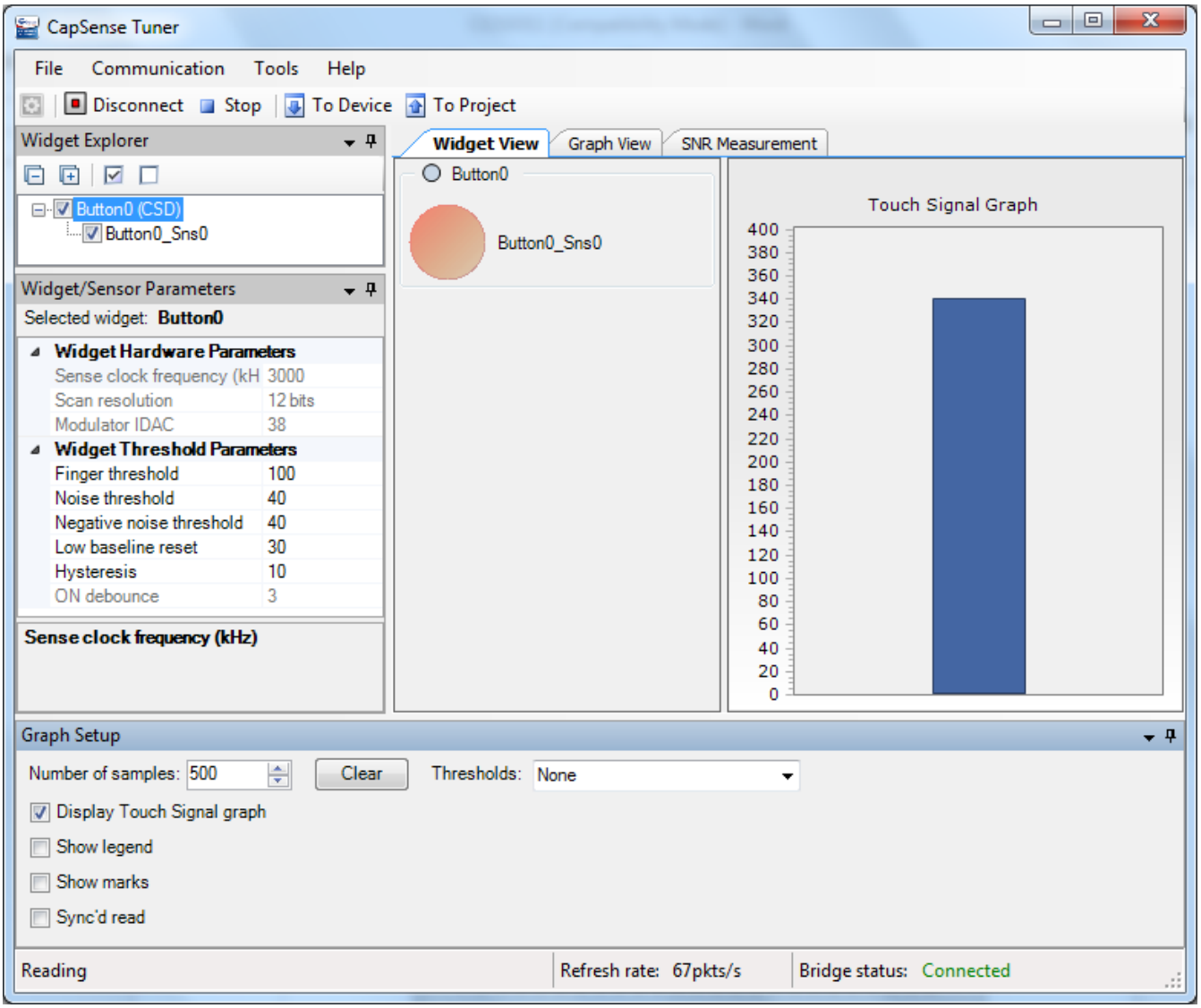
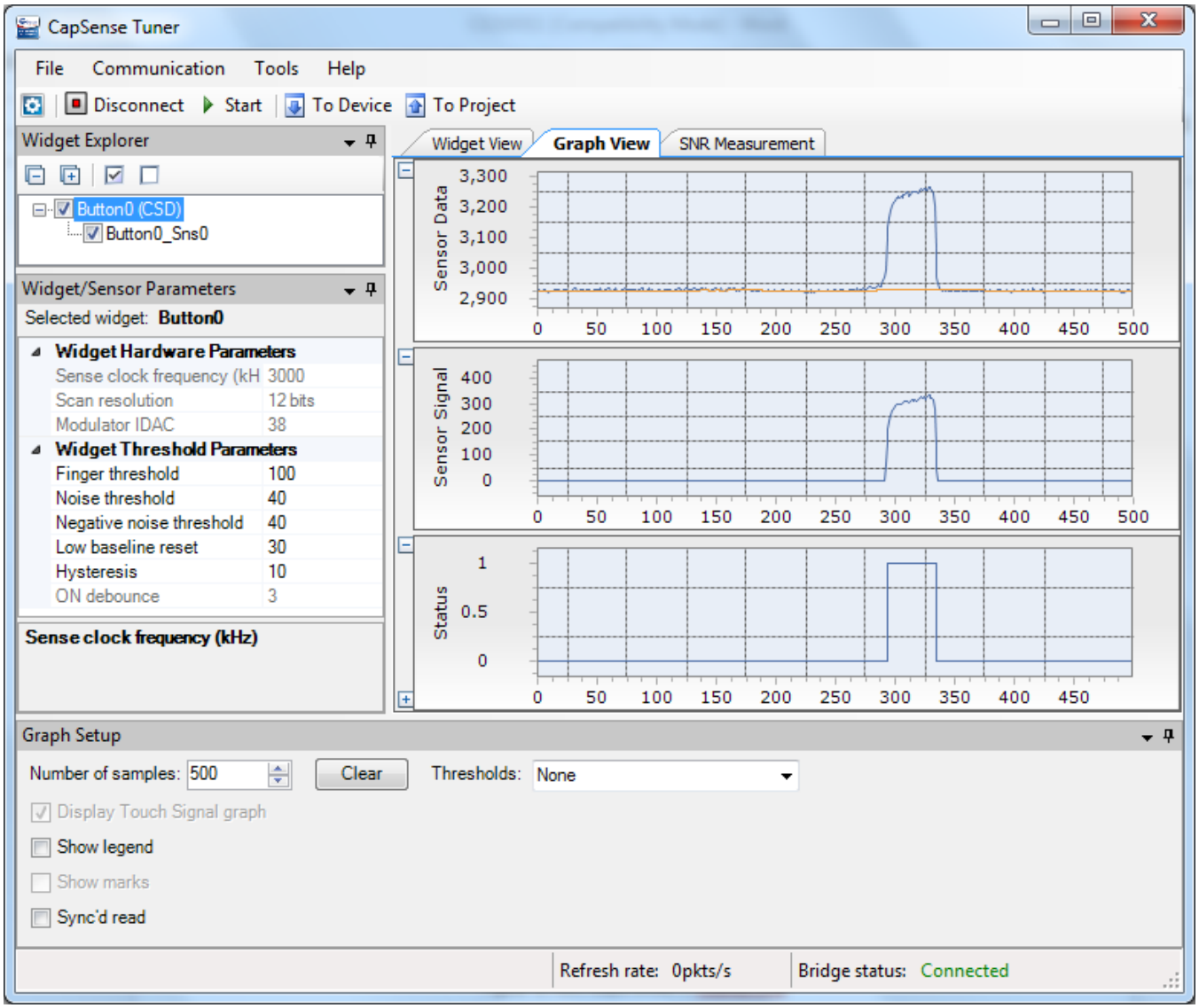
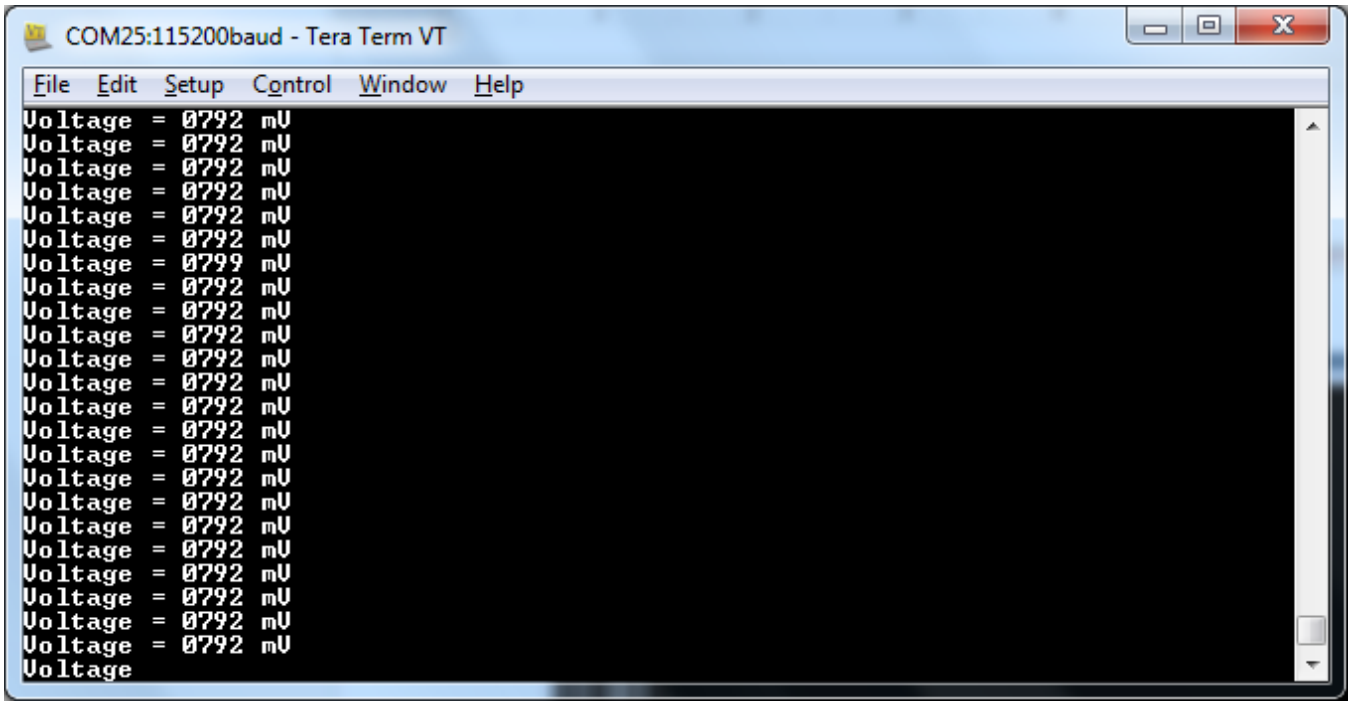


Figure 11. Tuner: Graph View



6) Use the potentiometer to vary the input to the CapSense_ADC Component. Observe how the LED brightness changes. You can also see the voltage reading in the HyperTerminal window (Figure 12).

Figure 12. ADC Output in HyperTerminal



Related Documents

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

Table 3. Related Documents

Application Notes		
AN85951	AN85951 - PSoC 4 CapSense Design Guide	Shows how to design capacitive touch sensing applications with the PSoC 4.
AN79953	Getting Started with PSoC 4	Describes the PSoC 4, and how to build a first PSoC Creator project.
Code Examples		
CE210311	PSoC 4 CapSense ADC Sequential	
PSoC Creator Component Datasheets		
CapSense_ADC	Supports various interfaces such as Button, Matrix Buttons, Slider, Touchpad, Proximity Sensor, and ADC.	
EZI2C	Supports one- or two-address decoding with independent memory buffers.	
Sw_Tx_UART	Supports 8-bit RS-232 data-format compliant serial transmitter.	
Pins	Supports connection of hardware resources to physical pins.	
Device Documentation		
PSoC 4 Datasheets	PSoC 4 Technical Reference Manuals	
Development Kit (DVK) Documentation		
CY8CKIT-041-40xx PSoC 4 S-Series Pioneer Kit		
CY8CKIT-041-41xx PSoC 4 S-Series 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:

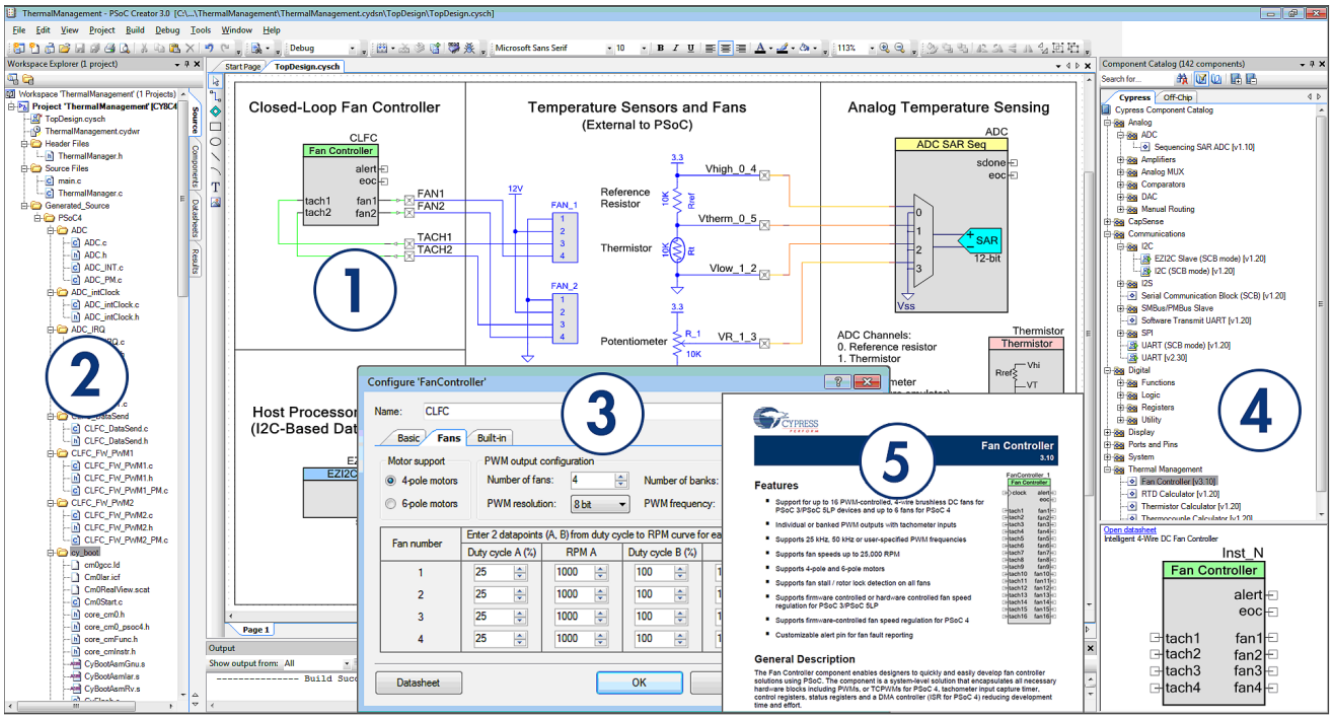
- **Overview:** PSoC Portfolio, PSoC Roadmap
- **Product Selectors:** PSoC 1, PSoC 3, PSoC 4, or PSoC 5LP. In addition, PSoC Creator includes a device selection tool.
- **Datasheets**: Describe and provide electrical specifications for the PSoC 4000, PSoC 4000S, PSoC 4100S, PSoC 4100, and PSoC 4200, 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.
- **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 13. 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 13. PSoC Creator Features



Document History

Document Title: CE210311 - CapSense® ADC Sequential

Document Number: 002-10311

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5144101	AKSM	02/19/2016	New code example

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Lighting & Power Control	cypress.com/powerpsoc
Memory	cypress.com/memory
PSoC	cypress.com/psoc
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless/RF	cypress.com/wireless

PSoC® Solutions

cypress.com/psoc

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Community | Forums | Blogs | Video | Training

Technical Support

cypress.com/support

PSoC and CapSense are registered trademarks and PSoC Creator is a trademark of Cypress Semiconductor Corp. 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

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 2016. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you under its copyright rights in the Software, a personal, non-exclusive, nontransferable license (without the right to sublicense) (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. Cypress also grants you a personal, non-exclusive, nontransferable, license (without the right to sublicense) 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 to the minimum extent that is necessary for you to exercise your rights under the copyright license granted in the previous sentence. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.
CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. 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 Company shall and hereby does release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. Company 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, 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.