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 the PSoC® 4 S-Series supports both capacitive sensing and input voltage measurement. The capacitive sensing mode can be used for touch sensing, and the ADC mode can be used for voltage measurement in applications such as temperature measurement. This code example shows how to scan a CapSense sensor and measure the voltage on any input pin in a time-multiplexed method using the CapSense_ADC Component.

This code example scans a single button sensor and measures the voltage across the potentiometer after the button sensor scanning is complete. The CapSense Tuner reads the button sensor data using I^2C communication. The ADC result is displayed on a PC via UART (SW_Tx_UART) communication. Depending on the ADC result, the brightness level of the LED is varied.

Requirements

Tool: PSoC Creator™ 3.3 SP2.4 and later versions

Programming Language: C (ARM® GCC 4.9.3, ARM MDK)

Associated Parts: All PSoC 4 S-Series Parts

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

Design

Figure 1 shows the PSoC Creator schematics of this code example. This example uses the CapSense_ADC, SW_Tx_UART, EZI2C Slave, PWM, and Pins Components.

The CapSense Component is configured with a one-button widget. The project uses the SmartSense™ (full Auto-tuning) method to quickly implement a single button sensor. The EZI2C Slave Component is used to monitor the sensor data on the PC using the CapSense Tuner software available in the PSoC Creator integrated design environment (IDE). The design uses the SW_Tx_UART Component to send the ADC data to the PC. The PWM Component modulates the LED brightness based on the output of the CapSense_ADC Component.

Figure 2 shows the firmware flow chart.

Figure 1. Component Schematics

CE210311 CapSense ADC Sequential

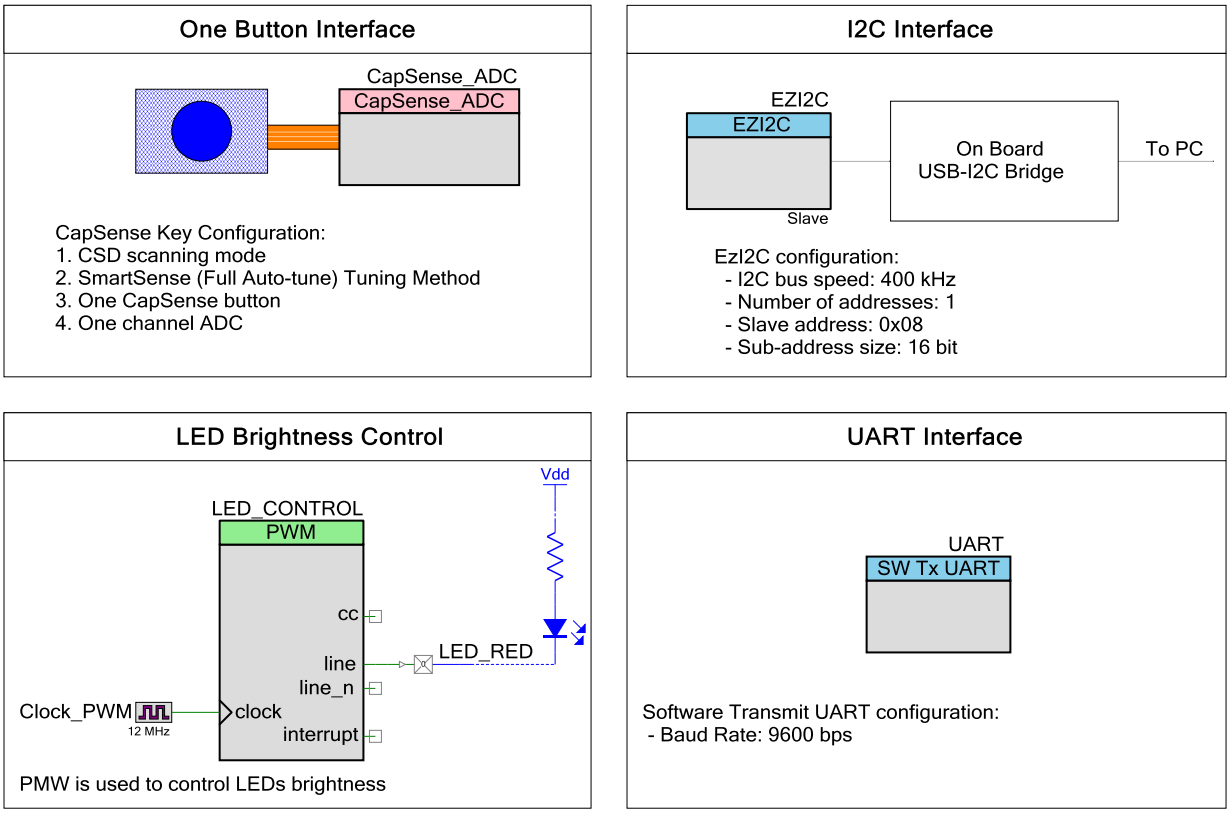
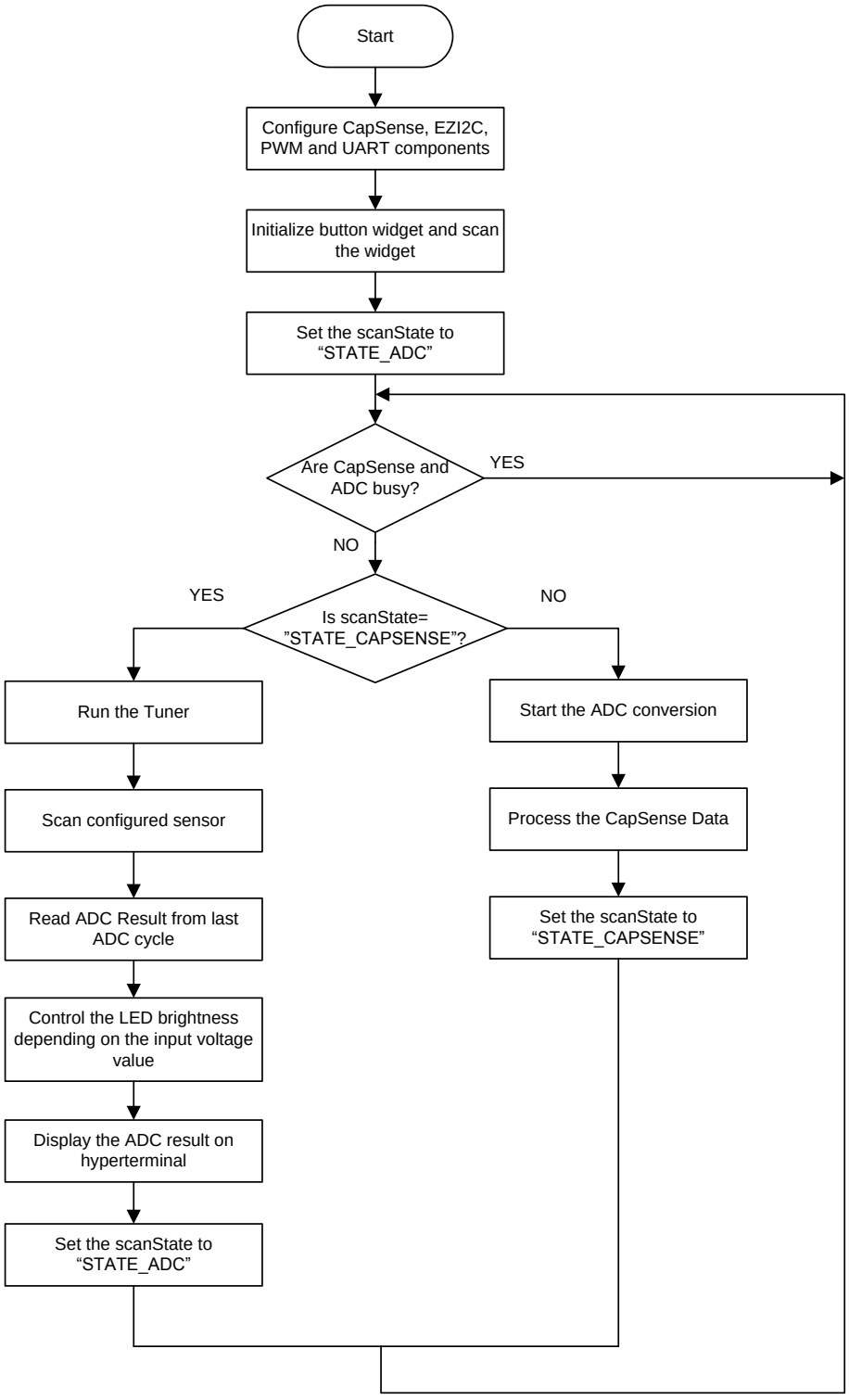


Figure 2. Firmware Flow Chart

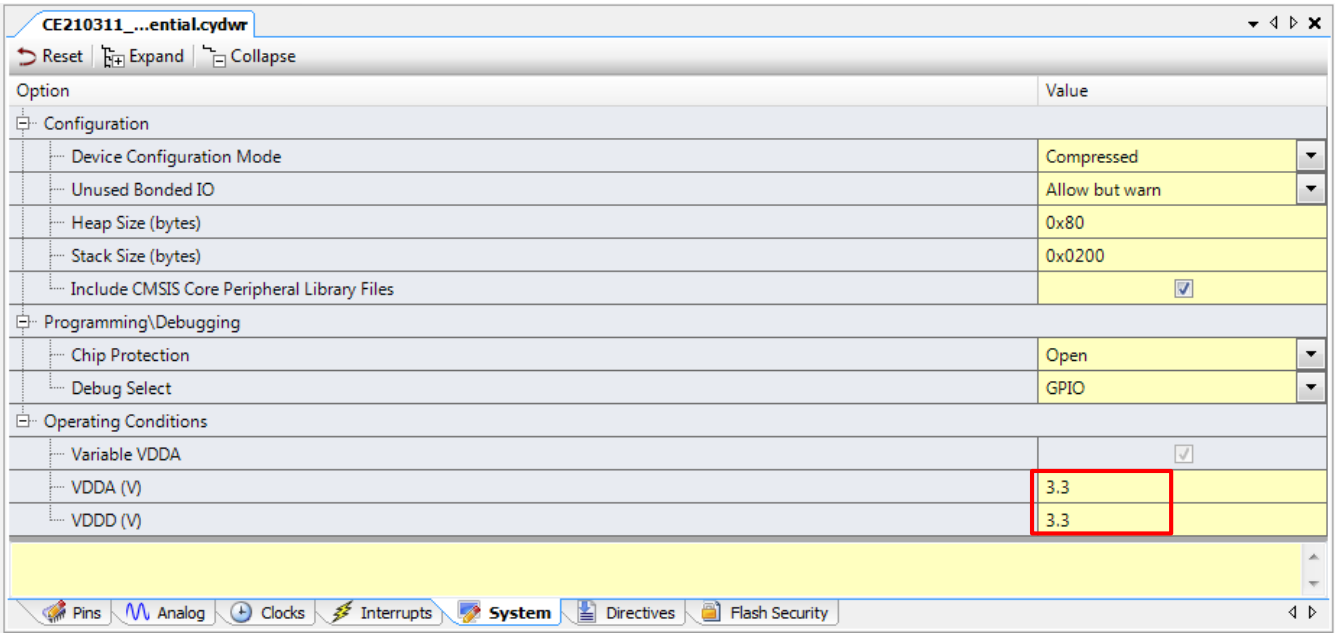


Design Considerations

This code example is designed to run on the CY8CKIT-041-40xx with the PSoC 4000S device. To port the design to other PSoC 4 devices and kits, you must change the target device in the Device Selector and change the pin assignments in the CYDWR settings.

The Macro VMAX_MV in *main.c* is set to the VDDA value specified in the CYDWR (Figure 3). This macro is used by the firmware to drive the LED. The VDDA value in the CYDWR should be set based on the operating voltage of the kit (1.9 V/3.3 V/5 V). By default, the VDDA is specified as 3.3 V.

Figure 3. CYDWR System Tab Settings



This code example works only on the devices that support fourth-generation CapSense. Refer to AN85951 – PSoC 4 CapSense Design Guide for a list of devices that support fourth-generation CapSense.

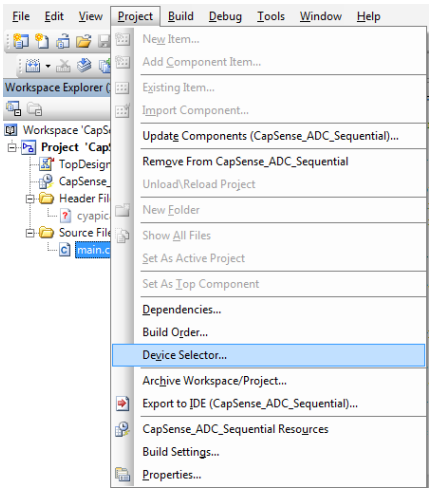
To switch from the CY8CKIT-041-40xx to other PSoC 4 pioneer kits, follow these steps:

1. Select the appropriate device with the Device Selector called from the project's context menu (Figure 4). Table 1 lists the device number for each pioneer kit.

   When you select the device, PSoC Creator assigns pins automatically for the selected device in the Design Wide Resources file.

   Note: If the assigned Component's pins are not as shown in Table 1 or you want to overwrite the existing pin assignments, double-click the project's Design Wide Resources file in the Workspace Explorer window and assign the pins. Refer to the device datasheet for information on pin assignments.

Figure 4. Select Device Selector from Project's Context Menu



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

Table 1. Pin Assignments for CapSense ADC Sequential Project

Pin Name	Development Kit	
	CY8CKIT-041-40xx (CY8C4045AZI-S423)	CY8CKIT-041-41xx (CY8C4146AZI-S433)
\CapSense_ADC :AdcInput\	P2[4]	P2[4]
\CapSense_ADC :Cmod\ (Cmod)	P4[1]	P4[1]
\CapSense_ADC :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. PSoC Creator Components

Component	Instance Name	Version	Hardware Resources
CapSense_ADC	CapSense_ADC	v3.0	CSD, 3* GPIO pins
EZI2C Slave (SCB mode)	EZI2C	v3.20	SCB, 2 GPIO pins
Digital Output Pin	LED_RED	v2.20	1 GPIO pin
PWM (TCPWM mode)	LED_CONTROL	v2.10	1 TCPWM
Clock	Clock_PWM	v2.20	1 clock divider
UART	SW_TX_UART	v1.50	1 GPIO pin

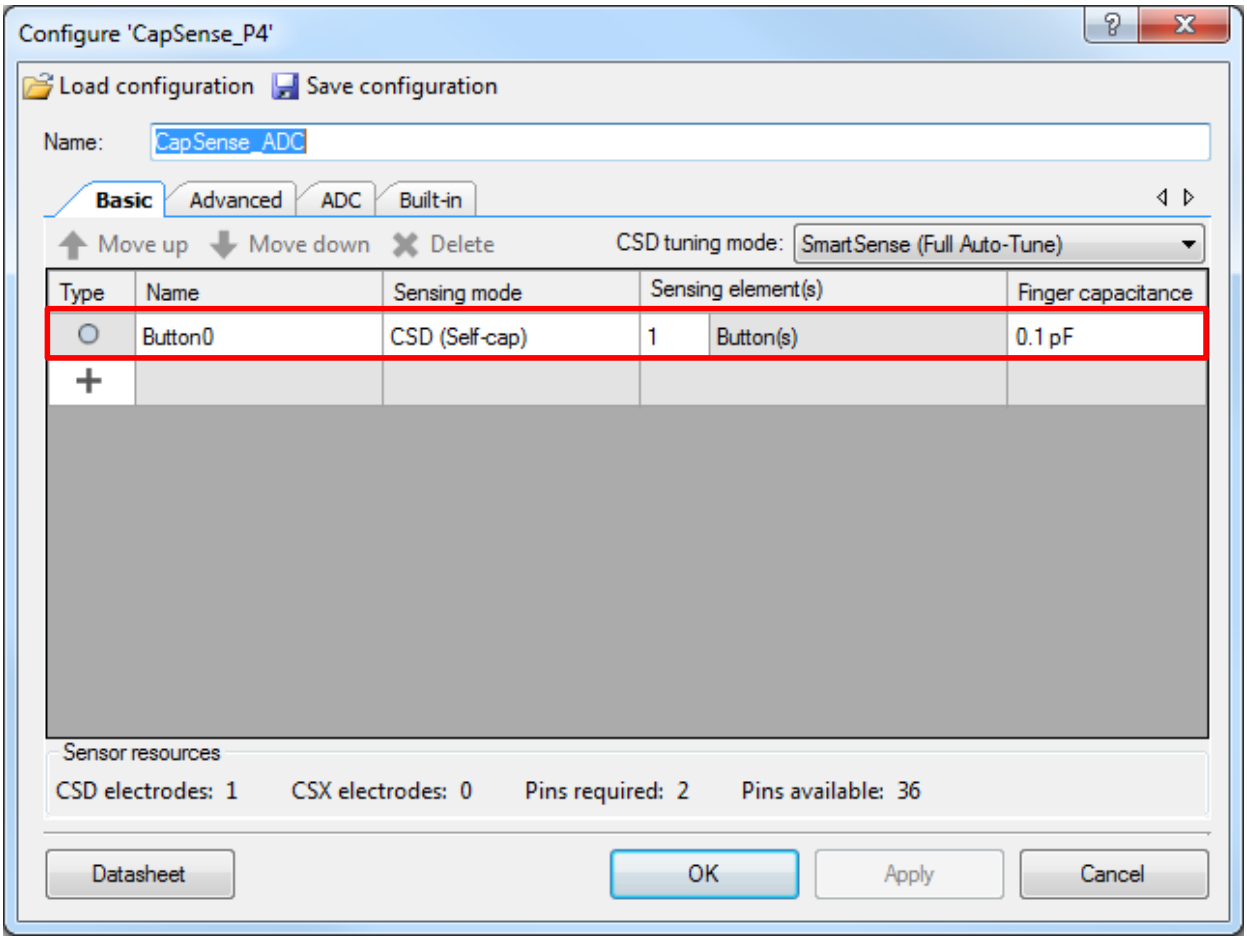
*The CapSense_ADC Component requires at least one sensor to be configured for proper operation. Hence, one pin is used for the sensor and another for CMOD connection.

Parameter Settings

CapSense ADC Component

Figure 5 shows the CapSense_ADC Component settings that are changed from their default values. See the CapSense ADC Component datasheet for additional information.

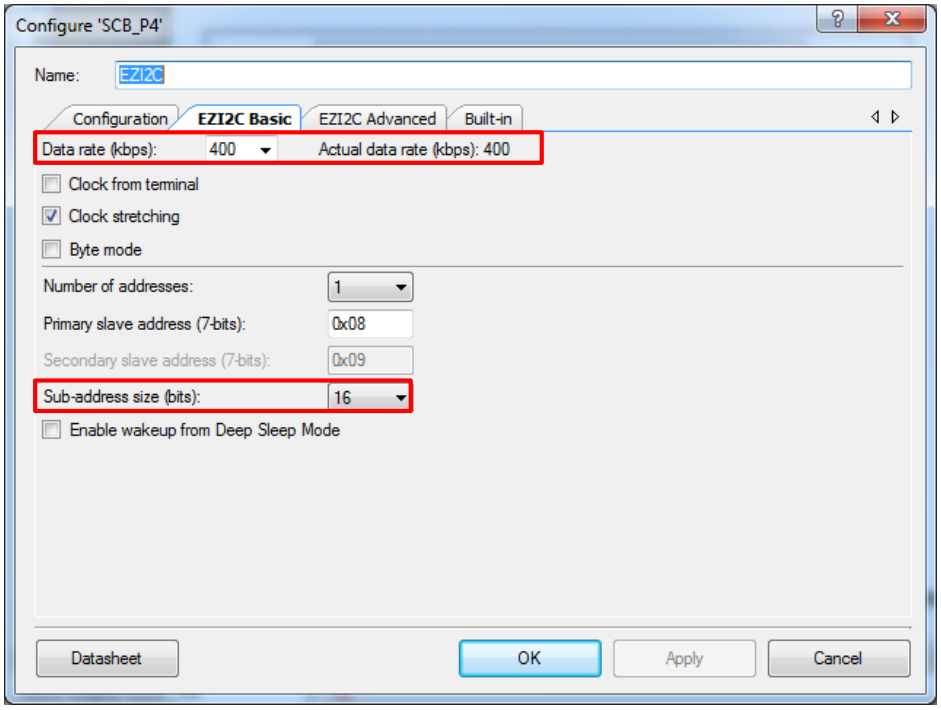
Figure 5. CapSense ADC Component Configuration – Enable at Least One Sensor



EZI2C Slave Component

Figure 6 shows the EZI2C Slave Component settings that are changed from their default values. See the SCB Component datasheet for additional information. Set a **Sub-address size** of 16 bits to allow the master to access data between 0 and 65535.

Figure 6. EZI2C Slave Component Basic Tab

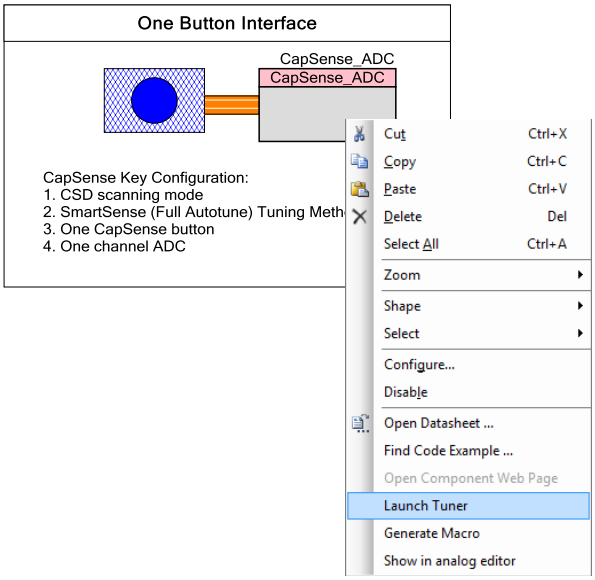


Operation

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

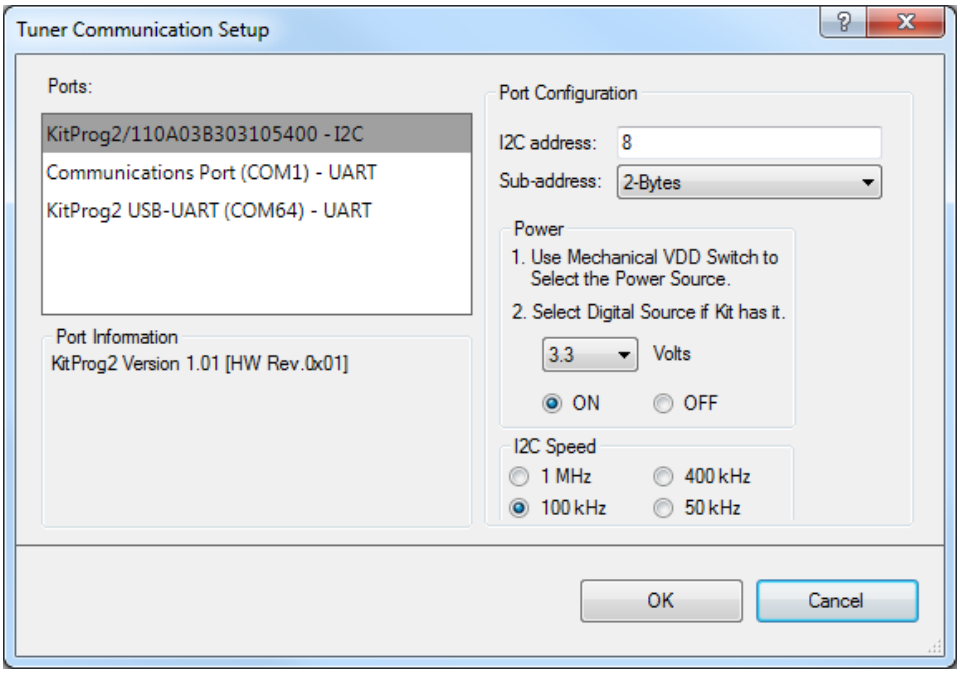
1. Launch the Tuner GUI. Right-click the CapSense Component, as shown in Figure 7. Select the **Launch Tuner** in the menu.

Figure 7. Launch Tuner GUI



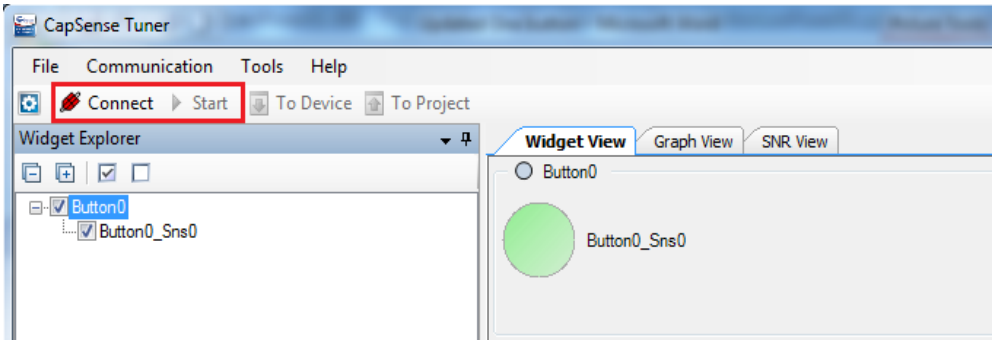
2. Navigate to **Tools** > **Protocol Settings** in the Tuner GUI menu to set up I²C communication (Figure 8).
   - Select the I²C device of the kit.
   - Configure the **I2C address, sub-address**, and **I2C speed,** as shown in Figure 8.

Figure 8. Setting Up I²C Communication



3. Press the **Connect** and then **Start** buttons (Figure 9).

Figure 9. Starting Communication



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

   The Tuner GUI displays the sensor debug data for all the sensing elements. Refer to Figure 10 and Figure 11.

5. Vary the input voltage by rotating the potentiometer and observe the LED brightness changes. You can also view the ADC result on the PC via the Tera Term window (Figure 12).

Figure 10. Tuner: Widget View

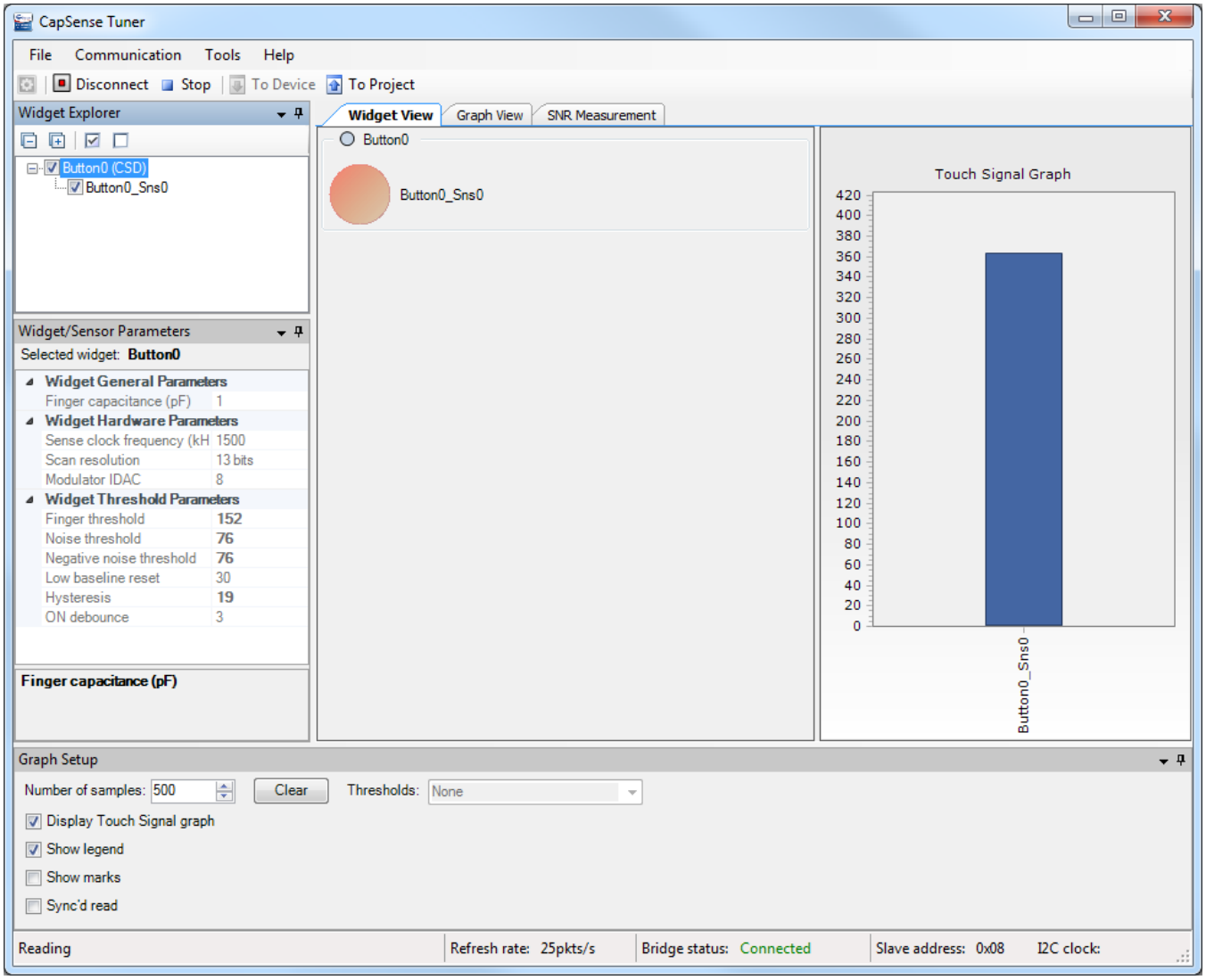


Figure 11. Tuner: Graph View

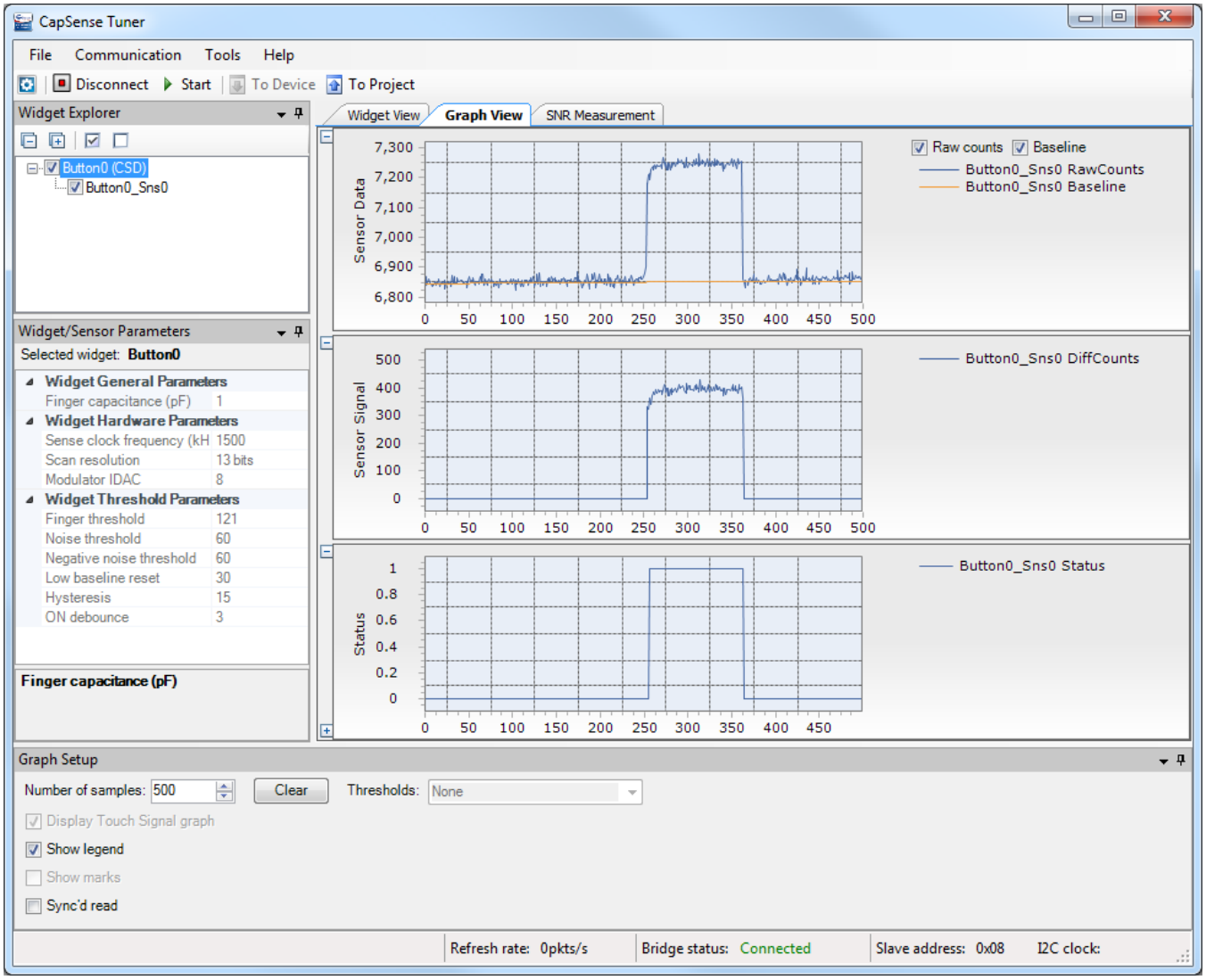
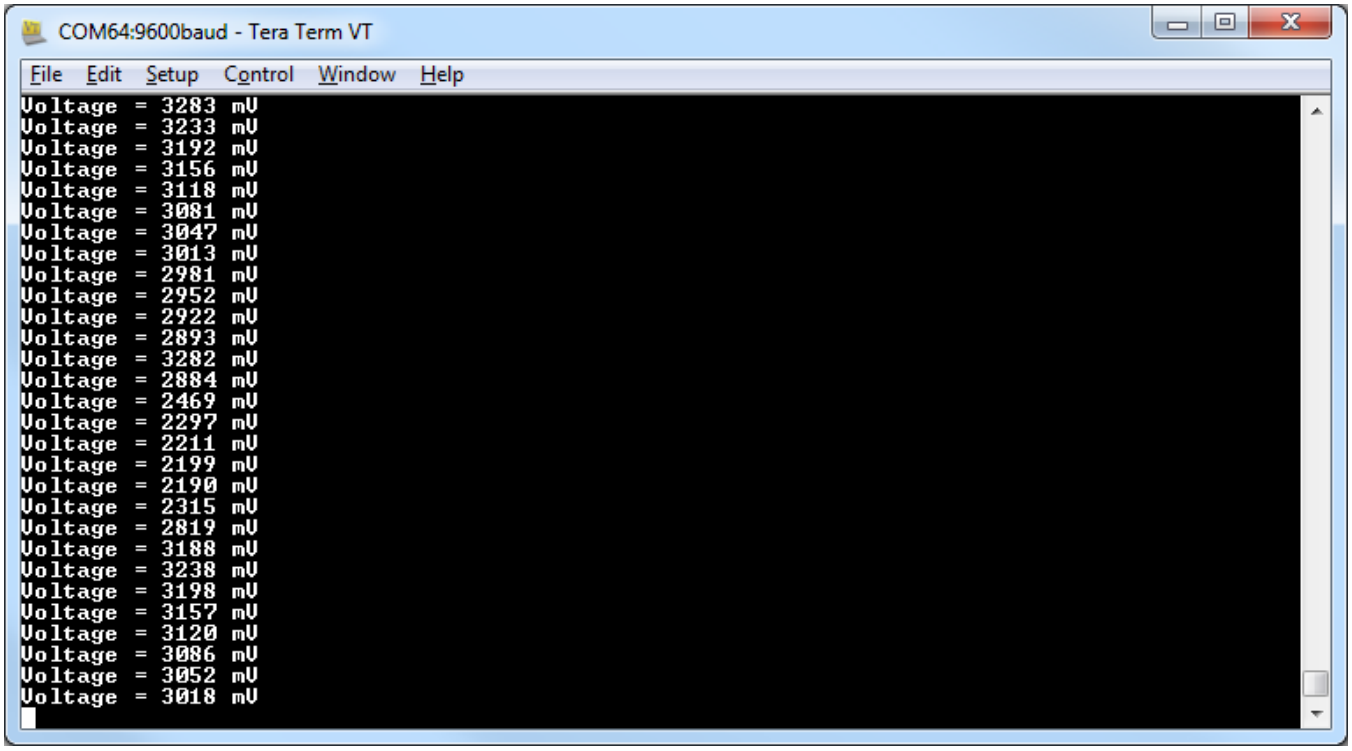


Figure 12. ADC Output Shown in Tera Term



Related Documents

Table 3 lists relevant application notes, PSoC Creator Component datasheets, and device and DVK documentation.

Table 3. Related Documents

Application Notes		
AN85951	PSoC 4 CapSense Design Guide	Shows how to design capacitive touch sensing applications with PSoC 4
AN79953	Getting Started with PSoC 4	Describes PSoC 4 and how to build a first PSoC Creator project
PSoC Creator Component Datasheets		
CapSense_ADC	Supports voltage measurement on any I/O pin	
EZI2C Slave	Supports I^2C slave operation	
Sw_Tx_UART	Supports 8-bit RS-232 data format–compliant serial transmitter	
PWM	Supports 16-bit fixed-function PWM implementation	
Pins	Supports connection of hardware resources to physical pins	
Clock	Supports local clock generation	
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 to select the right PSoC device for your design 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:

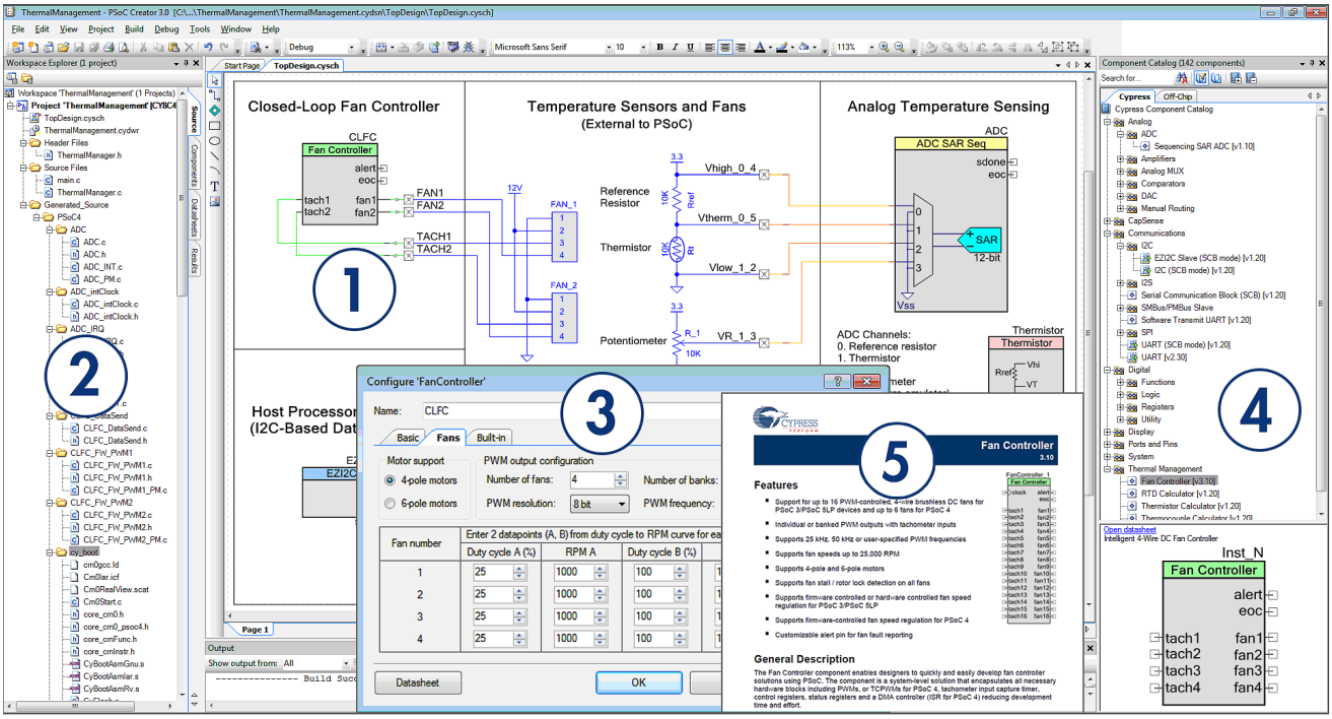
- **Overview:** PSoC Portfolio, PSoC Roadmap
- **Product Selectors:** PSoC 1, PSoC 3, PSoC 4, or PSoC 5LP. In addition, PSoC Creator includes a device selection tool.
- **Datasheets:** Describe and provide electrical specifications for the PSoC 3, PSoC 4, and PSoC 5LP device families.
- **CapSense Design Guides:** Learn how to design capacitive touch-sensing applications with the PSoC 3, PSoC 4, and PSoC 5LP families 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.
- **Technical Reference Manuals (TRM):** Provide detailed descriptions of the architecture and registers in each of the PSoC 3, PSoC 4, and PSoC 5LP device families.
- **PSoC Training Videos**: These videos provide step-by-step instructions on getting started building complex designs with PSoC.
- **Development Kits:**
  - CY8CKIT-041 PSoC 4 S-Series Pioneer Kit is an easy-to-use and inexpensive development platform. This kit includes connectors for Arduino™ compatible shields and Digilent® Pmod™ daughter cards.
  - CY8CKIT-145 is a very low-cost prototyping platform for evaluating the PSoC 4 S-Series of devices.
- The MiniProg3 device provides an interface for flash programming and debugging.

PSoC Creator

PSoC Creator is a free Windows-based 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
*A	5410758	DCHE	08/22/2016	Updated all the figures to match production version of CapSense 3.0 Component. Updated template

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

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

Cypress Developer Community

Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

PSoC is a registered trademark and PSoC Creator is a trademark of Cypress Semiconductor Corporation. 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 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.