CE214023 – CapSense® Proximity Sensing

Objective

This code example demonstrates how to implement low-power CapSense® proximity sensing with a proximity-sensing distance of 5 cm at an average current consumption of 25 µA.

Overview

This code example demonstrates CapSense-based proximity sensing using a PCB trace as a proximity sensor. A proximity-sensing distance of 5 cm is achieved using a rectangular loop sensor with a 9-cm diagonal. Proximity detection is indicated by controlling the brightness of an LED. The LED has a minimum brightness when the hand is at a distance of 5 cm; it gradually increases as the hand approaches the sensor. Using the low-power modes available in the PSoC® 4100S device, an average current of 25 µA is achieved while detecting the proximity of a hand at 5 cm.

Requirements

Tool: PSoC Creator™ 4.0 or later versions

Programming Language: C (ARM® GCC 4.9.3)

Associated Parts: All PSoC 4100S parts

Related Hardware: CY8CKIT-041-41XX PSoC 4100S Pioneer Kit

Design

Figure 1 shows the PSoC Creator schematic of this code example. It uses the CapSense, EZI2C Slave, PWM, Clock, and Pins Components.

The CapSense Component is configured to scan a proximity sensor. The EZI2C Slave Component is used to monitor the sensor data on a PC using the CapSense Tuner available in the PSoC Creator integrated design environment (IDE). The PWM Component controls the brightness of the onboard Red LED by driving a pseudo-random PWM signal.

Figure 2 shows the flow chart for the code example. To reduce the power consumed by the PSoC device and to detect the proximity of a fast-approaching hand, this code example implements two power modes: Fast Scan and Slow Scan. When proximity is detected, the PSoC device is in the Fast Scan mode, and when the user is away from the sensor for a specific duration, the Slow Scan mode is used.

In the Fast Scan mode, the proximity sensor is scanned at a refresh rate of 50 Hz (or a scan interval of 20 ms), and the Red LED is driven based on the proximity of the user's hand with respect to the kit. The PSoC device is put into the CPU Sleep mode after the CapSense data is processed and the LED is driven. The watchdog timer is used to periodically wake up the device from the Sleep mode. This mode provides an optimum touch response, but consumes a higher average current compared to the Slow Scan mode.

In the Slow Scan mode, the proximity sensor is scanned at a refresh rate of 5 Hz (or a scan interval of 200 ms). The LED is turned OFF, and the PSoC device is put into the Deep Sleep mode periodically. The Slow Scan mode consumes a lower average current of 25 µA, but with a slower proximity detection response. Once proximity is detected in the Slow Scan mode, the PSoC device switches to the Fast Scan mode to provide a fast proximity detection response at the expense of a higher average current consumption.

Figure 1. TopDesign

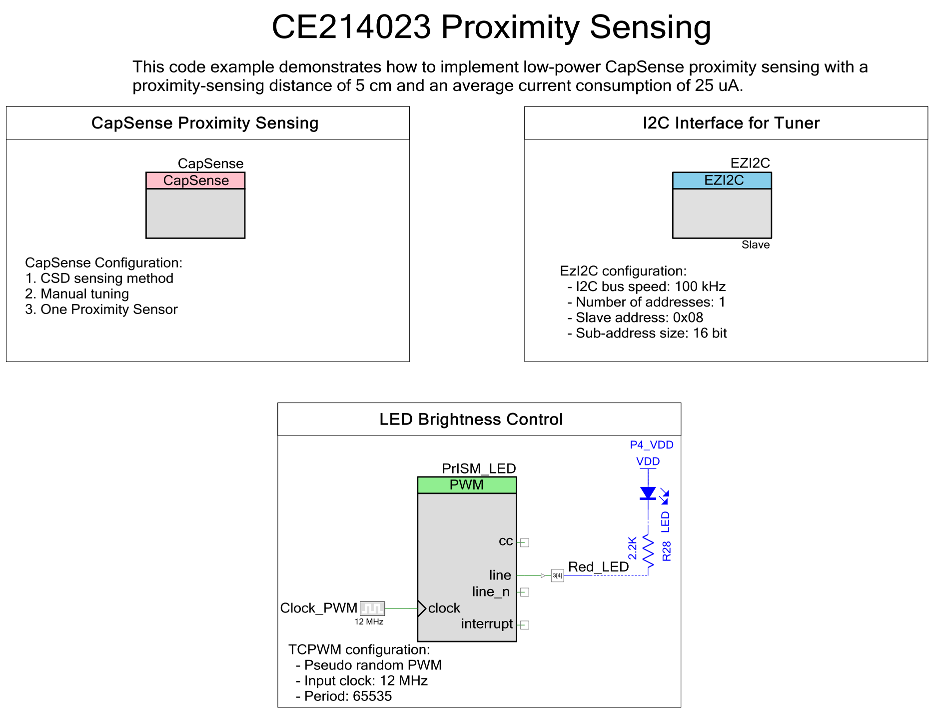
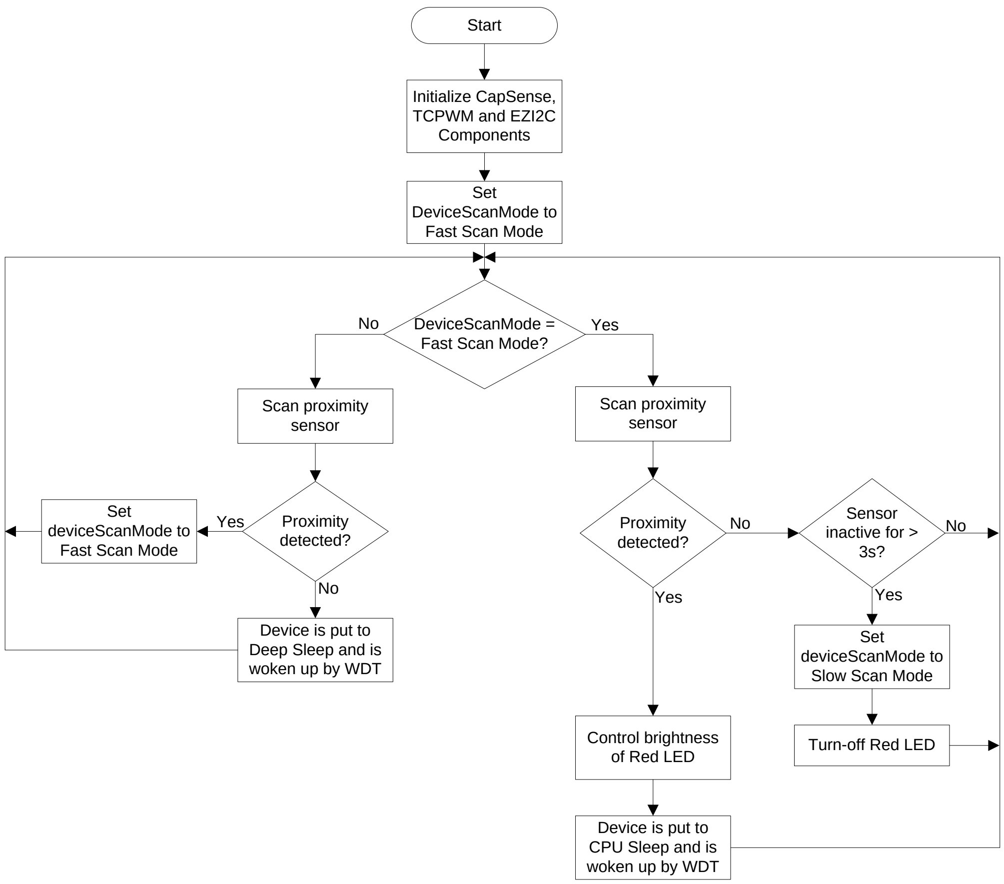


Figure 2. Firmware Flow Chart



Design Considerations

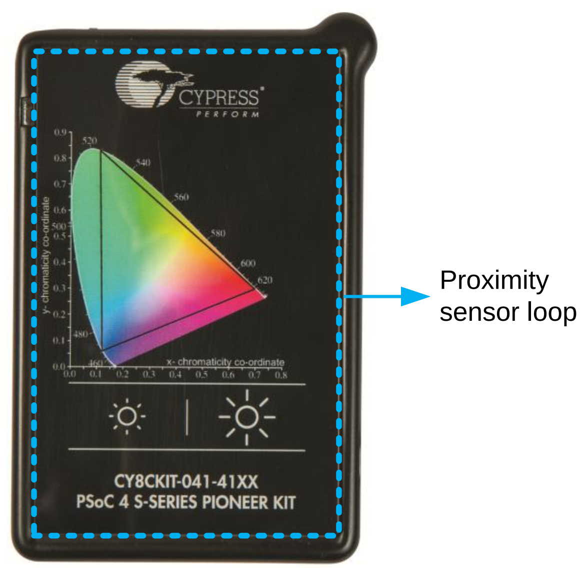
This code example is designed to run on the CY8CKIT-041-41XX PSoC 4100S Pioneer Kit with the PSoC 4100S device. To port the design to other PSoC 4 devices and kits, you must change the target device in the Device Selector, change the pin assignments in the *.cydwr* settings, and retune the CapSense sensors. For the tuning procedure, see AN85951 – PSoC 4 and PSoC Analog Coprocessor CapSense Design Guide and AN92239 – Proximity Sensing with CapSense.

The proximity sensing distance reduces to ~1 cm when the device is powered from the onboard battery. This is due to the reduced ground coupling between human body and device ground. To increase ground coupling, increase the ground plane area by attaching a conductive metal to the ground pin on the PSoC 4100S Pioneer board.

Hardware Setup

Figure 3 shows the proximity sensor loop location on the CY8CKIT-041-41XX PSoC 4100S Pioneer Kit. The code example works with the default settings on CY8CKIT-041-41XX. If the settings are different from the default values, see the “Switches Default Position” table in the kit guide to learn how to reset to default settings.

Figure 3 Proximity Sensor Loop on CY8CKIT-041-41XX



Software Setup

This code example does not require any special software considerations.

Note: When the kit is battery-powered and if EZ-BLE PRoC™ Module is active, the power supply to PSoC 4000S is not stable. This causes noise on the proximity sensor and therefore it is recommended to erase the EZ-BLE PRoC Module flash before testing this code example.

PSoC Creator Components

Table 1 lists the PSoC Creator Components used in this example, as well as the hardware resources used by each of the Components.

Table 1. PSoC Creator Components

Component	Instance Name	Version	Hardware Resources
CapSense	CapSense	v3.10	CSD, 3 GPIOs
EZI2C Slave (SCB mode)	EZI2C	v3.20	SCB, 2 GPIOs
Clock	Clock_PWM	v2.20	Clock Divider

PWM (TCPWM mode)	PrISM_LED	v2.10	TCPWM
Digital Output Pin	Red_LED	v2.20	1 GPIO pin

Parameter Settings

CapSense

Figure 4, Figure 5, Figure 6, and Figure 7 show the CapSense Component settings that are changed from the default values. See the CapSense Component datasheet for additional information.

Figure 4. CapSense Component – Basic Tab Configuration

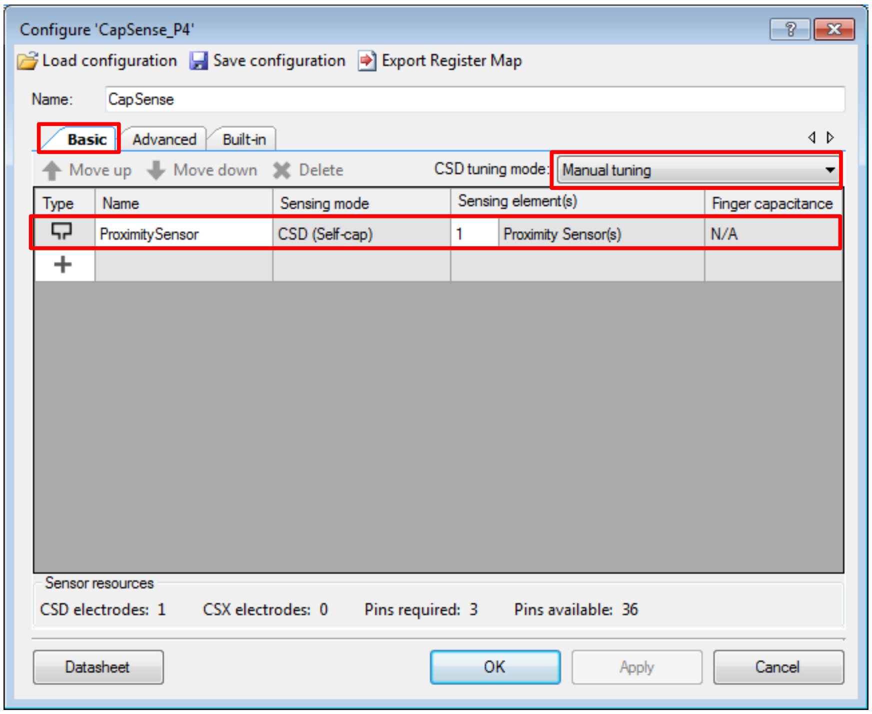


Figure 5. CapSense Component – Advanced Tab General Settings

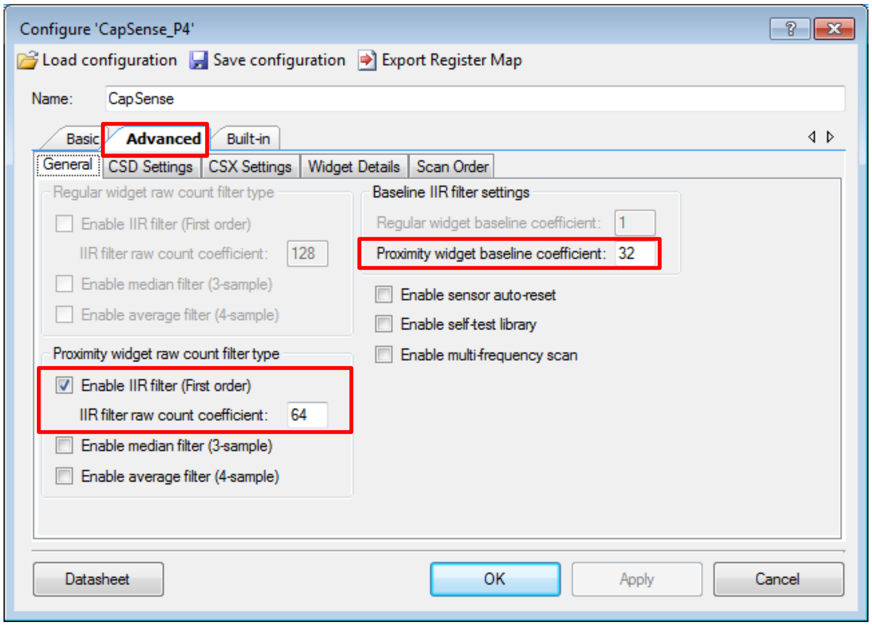


Figure 6. CapSense Component – Advanced Tab CSD Settings

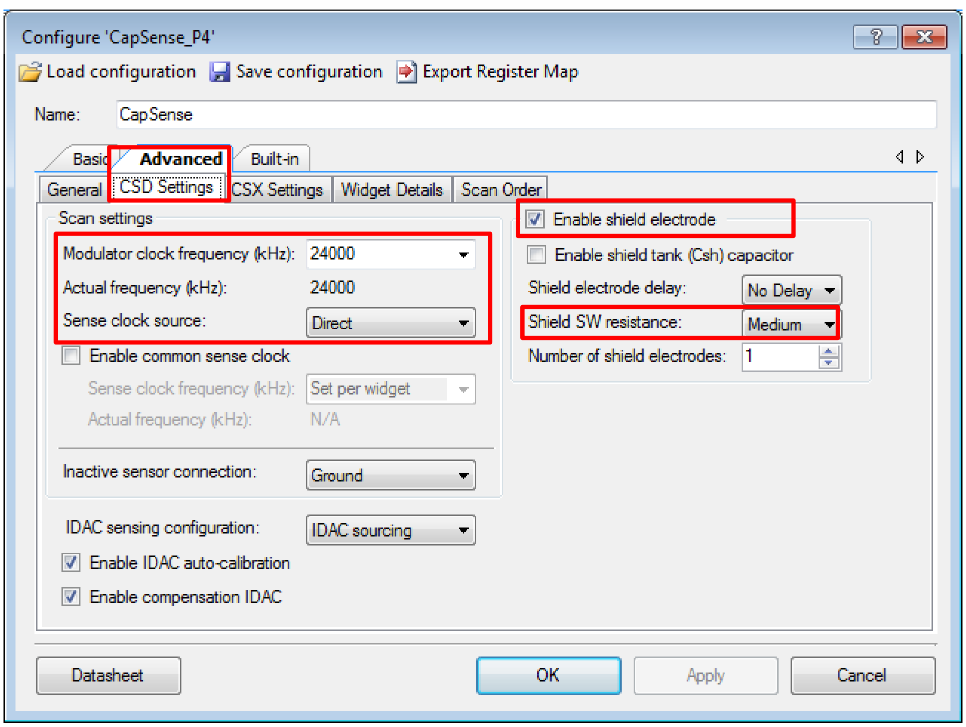
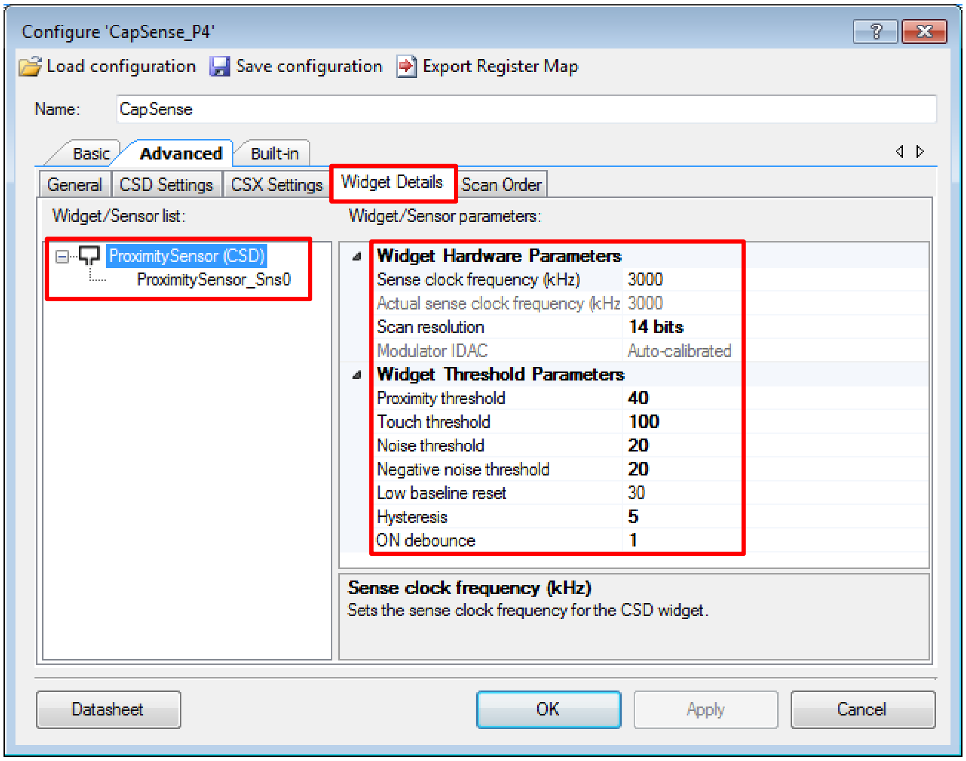


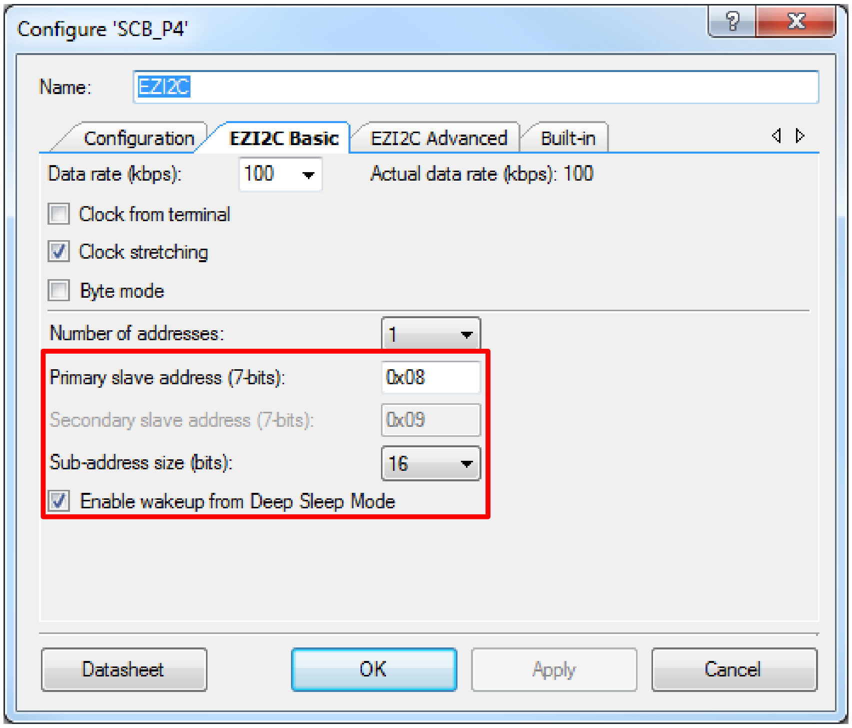
Figure 7. CapSense Component – Advanced Tab Widget Details Settings



EZI2C Slave

Figure 8 shows the non-default EZI2C Slave Component settings. See the SCB Component datasheet for additional information.

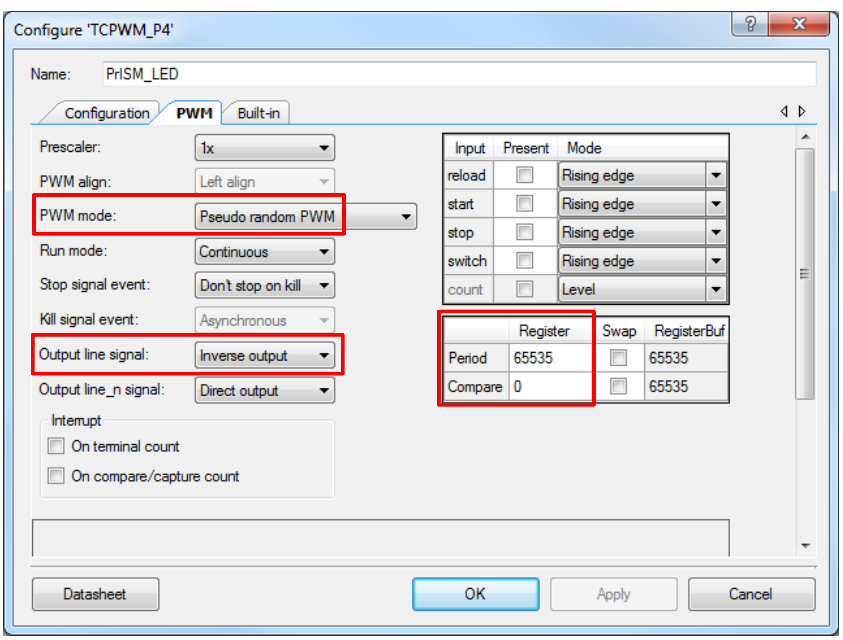
Figure 8. EZI2C Component Basic Settings



PWM

Figure 9 shows the non-default PWM Component settings. See the TCPWM Component datasheet for additional information.

Figure 9. PWM Component Configuration



Design-Wide Resources

Figure 10 and Figure 11 show the non-default .cydwr settings for the project.

Figure 10. .cydwr Pins Tab Settings

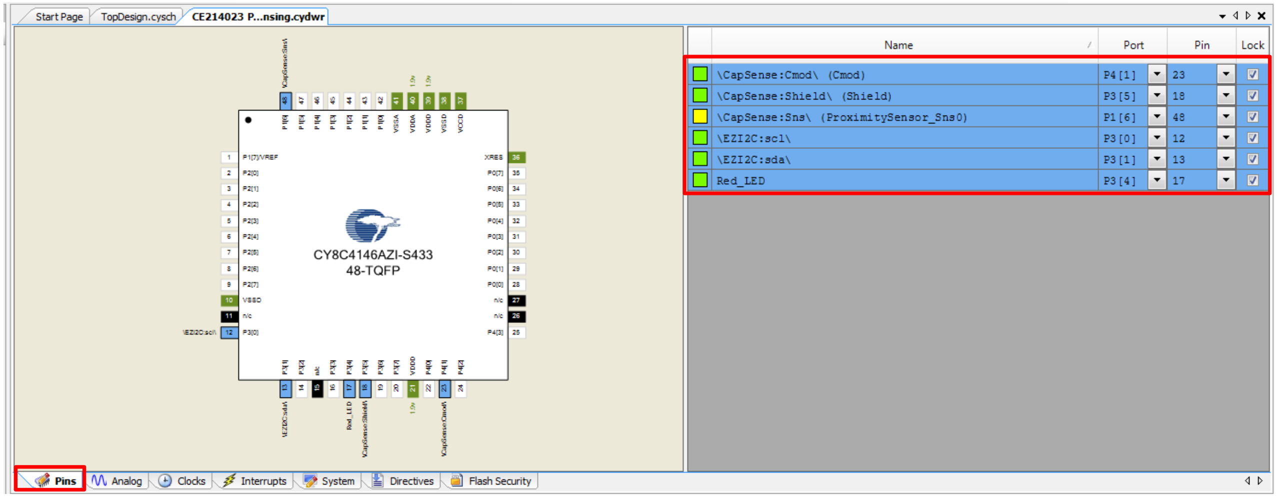
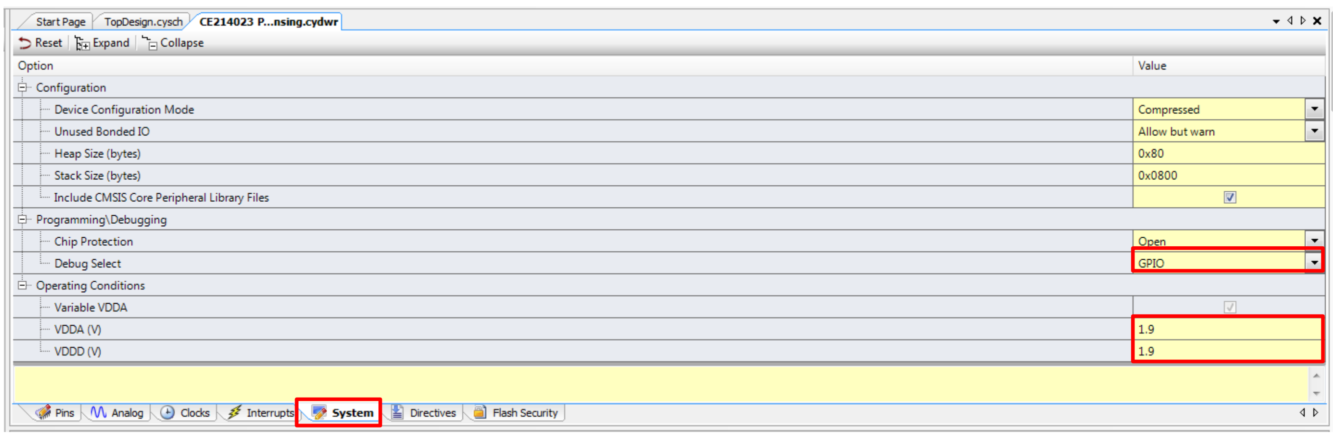


Figure 11. .cydwr System Tab Settings



Note: For a PSoC 4100S device, the CapSense V_{REF} voltage is set based on the VDDA setting in the .cydwr tab per Table 2.

Table 2. CapSense V_{REF} Values Based on VDDA Setting

VDDA (V)	V_{REF} (V)
< 2.7	1.2
2.7 to 4.8	2.1
≥ 4.8	4.2

If VDDA is set to 1.9 V in the .cydwr tab, V_{REF} is set to 1.2 V. This V_{REF} voltage ensures that the CapSense tuning parameters do not vary with respect to VDDA, thereby avoiding retuning of the sensors.

Operation

Follow these steps to test the project:

1. Select the *CE214023 Proximity Sensing.cywrk* file on the PSoC Creator Start Page at **Examples and Kits** > **Kits** > **CY8CKIT-041-41XX**. Select a location to save the code example.
2. Build the project (**Build** > **CE214023 Proximity Sensing**).
3. Connect the PSoC 4100S Pioneer Kit to your computer using the USB cable provided.

4. Program the PSoC 4100S device (**Debug** > **Program**). See the kit guide for details on programming the kit.
5. Bring your hand towards the kit and notice that at a distance of about 5 cm, the Red LED turns ON with the minimum brightness.

   Note: The proximity sensing distance is ~1 cm when the kit is powered from the onboard battery. See the Design Considerations section for the details on how to improve the proximity sensing distance for battery-powered applications.
6. Bring your hand closer and notice that the LED brightness increases as the distance between your hand and the kit is reduced.
7. Connect an ammeter between the P4.VDD and VDD test points on the main board to measure the PSoC 4100S current consumption. See the "Current Measurement Switch" section in the kit guide for complete details on power measurement steps.
8. Remove your hand from the proximity sensor, wait for three seconds, and notice that the average current is approximately 25 µA in the Slow Scan mode.

   Note: At 5 V, the average current consumption is much higher than 25 µA. This is because the VDDA value in the *.cydwr* settings is set to 1.9 V instead of the actual operating voltage. See the "Low Voltage Analog Boost Clocks" section in the PSoC 4 System Reference Guide for more information.
9. Bring your hand towards the kit and notice that the average current increases to approximately 3 mA because the device is in the Fast Scan mode and the LED is turned ON.
10. If high noise is observed or false triggering is observed at 1.8 V, do the following for a proper operation:
    a. Enable the "shield tank (Csh) capacitor" option in **CapSense component** > **Advanced tab** > **CSD settings.**
    b. In the .**cydwr pins tab**, assign the Cshield_tank pin to P4[3].
    c. In kit hardware, change the C12 capacitor value to 10 nF.

Upgrade Information

The code example is updated to the latest version of PSoC Creator and therefore does not require an upgrade.

Related Documents

Table 3 lists the relevant application notes, code examples, PSoC Creator Component datasheets, device documentation, and development kit (DVK) documentation.

Table 3. Related Documents

Application Notes		
AN79953	Getting Started with PSoC 4	Describes PSoC 4 and how to build your first PSoC Creator project
AN85951	PSoC 4 and PSoC Analog Coprocessor CapSense Design Guide	Describes PSoC 4 and PSoC Analog Coprocessor CapSense Component tuning
AN92239	Proximity Sensing with CapSense	Describes how to implement capacitive proximity-sensing applications using PSoC CapSense
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
PWM	Supports 16-bit fixed-function pseudo-random PWM implementation
EZI2C Slave	Supports I^2C slave operation
Pins	Supports connection of hardware resources to physical pins
Clock	Supports local clock generation
Device Documentation	
PSoC 4100S Family Datasheet	
PSoC 4100S Family PSoC 4 Architecture Technical Reference Manual	
Development Kit (DVK) Documentation	
CY8CKIT-041-41XX PSoC 4100S 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:

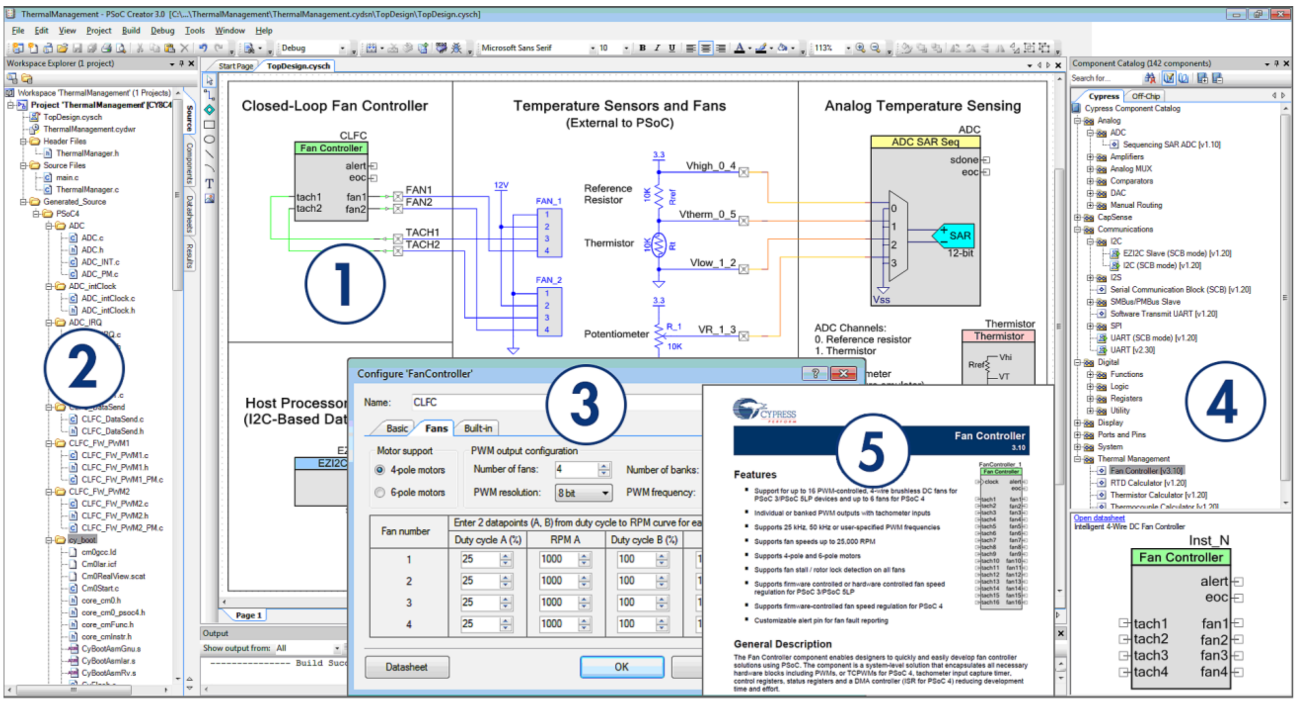
- **Overview:** PSoC Portfolio, PSoC Roadmap
- **Product Selectors:** PSoC 1, PSoC 3, PSoC 4, or PSoC 5LP. In addition, PSoC Creator includes a Device Selector tool.
- **Datasheets** describe and provide electrical specifications for the PSoC 3, PSoC 4, and PSoC 5LP device families.
- **CapSense Design 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 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-041-41XX PSoC 4100S Pioneer Kit is an easy-to-use and inexpensive development platform. This kit includes connectors for Arduino™ compatible shields.
  - CY8CKIT-145 is a very low-cost prototyping platform for evaluating PSoC 4 S-Series 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 12. With PSoC Creator, you can:

1. Drag and drop Components to build your hardware system design in the main design workspace
2. Co-design 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 12. PSoC Creator Features



Document History

Document Title: CE214023 – CapSense® Proximity Sensing

Document Number: 002-14023

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5401366	SRDS	11/21/2016	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
Internet of Things	cypress.com/iot
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