CE216892 - USB HID Trackpad

Objective

This code example implements a CapSense® trackpad and interfaces the trackpad as a USB Mouse Human Interface Device (HID) to a Windows PC.

Overview

This code example implements a CapSense trackpad and two button sensors using a PSoC 4100S device. The PSoC device is interfaced to a Windows PC as a mouse using the USB HID protocol.

The trackpad controls the cursor on the PC and the two button sensors act as right-click and left-click buttons. To reduce the power consumed by the PSoC device and provide an optimum touch response, this code example implements two modes: Fast Scan and Slow Scan.

Figure 1. Trackpad and Button Sensors on CY8CKIT-041-41XX

1. CapSense trackpad	2. CapSense buttons

Note: The above figure shows the kit without the Overlay

Requirements

Tool: PSoC Creator™ 4.0 and 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 2 shows the PSoC Creator schematic of this code example. This code example uses the CapSense, EZI2C Slave, and Pin Components.

The CapSense Component is configured to scan a 7x7 trackpad widget, two button widgets, and one ganged widget. Trackpad touch coordinates are used to the control the position of the mouse cursor on the PC. The two button widgets are used for left-click and right-click operations. The ganged sensor is a combination of all the column sensors of the trackpad widget and the two button widgets. The ganged sensor is used to implement the wake-on-approach feature; that is, the device detects an approaching finger at a distance of 2 cm from the trackpad. The Red LED glows when either a proximity or touch event is detected.

The EZI2C Slave Component is used to transfer the trackpad and button sensor data to the onboard KitProg2. KitProg2 acts as an I^2C-USB HID bridge. It reads the trackpad and button sensor data from the PSoC 4100S device via the I^2C interface and interfaces to the PC as a USB HID Mouse. The I^2C Component is also used for CapSense tuning. When tuning is enabled, the HID mouse functionality is disabled.

To optimize device power consumption and provide an optimum touch response, this code example implements two power modes: Fast Scan mode and Slow Scan mode. When the user is interacting with the trackpad or button sensors, the device is in the Fast Scan mode; when the sensors are inactive for a specific duration, the Slow Scan mode is used. In the Fast Scan mode, all the trackpad sensors and button sensors are scanned at a refresh rate of 50 Hz and the I^2C buffer is updated with trackpad XY coordinates and the button sensor status. This mode provides an optimum touch response but consumes higher power when compared to the Slow Scan mode.

In the Slow Scan mode, the trackpad column sensors and two button sensors are ganged and scanned as a single sensor at a refresh rate of 6 Hz. The Slow Scan mode consumes a lower average power but provides a slower touch response. The watchdog timer is used to periodically wake up the device from the deep-sleep mode at a refresh rate of 50 Hz in the Fast Scan mode and 6 Hz in the Slow Scan mode.

Figure 2. Top Design – CapSense and EZI2C

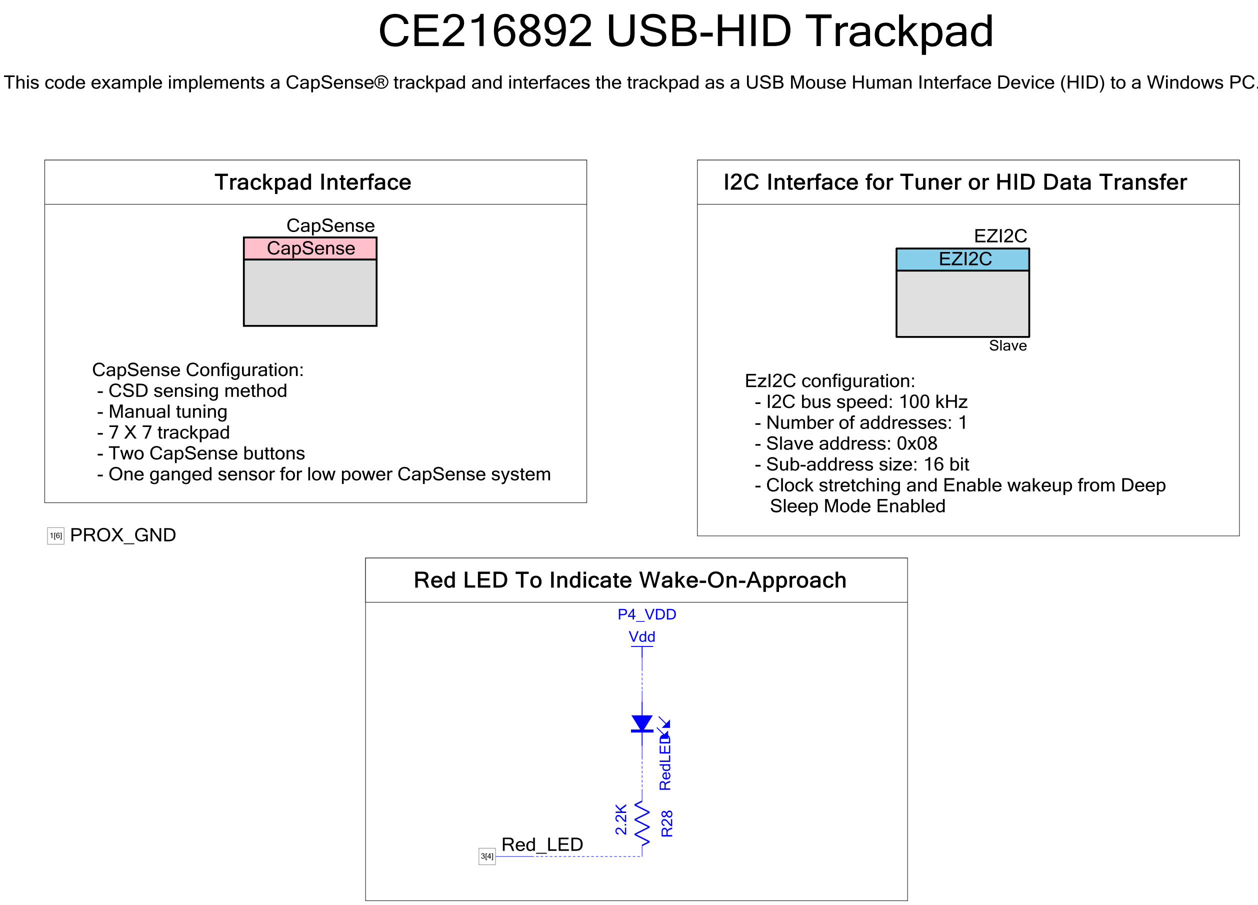
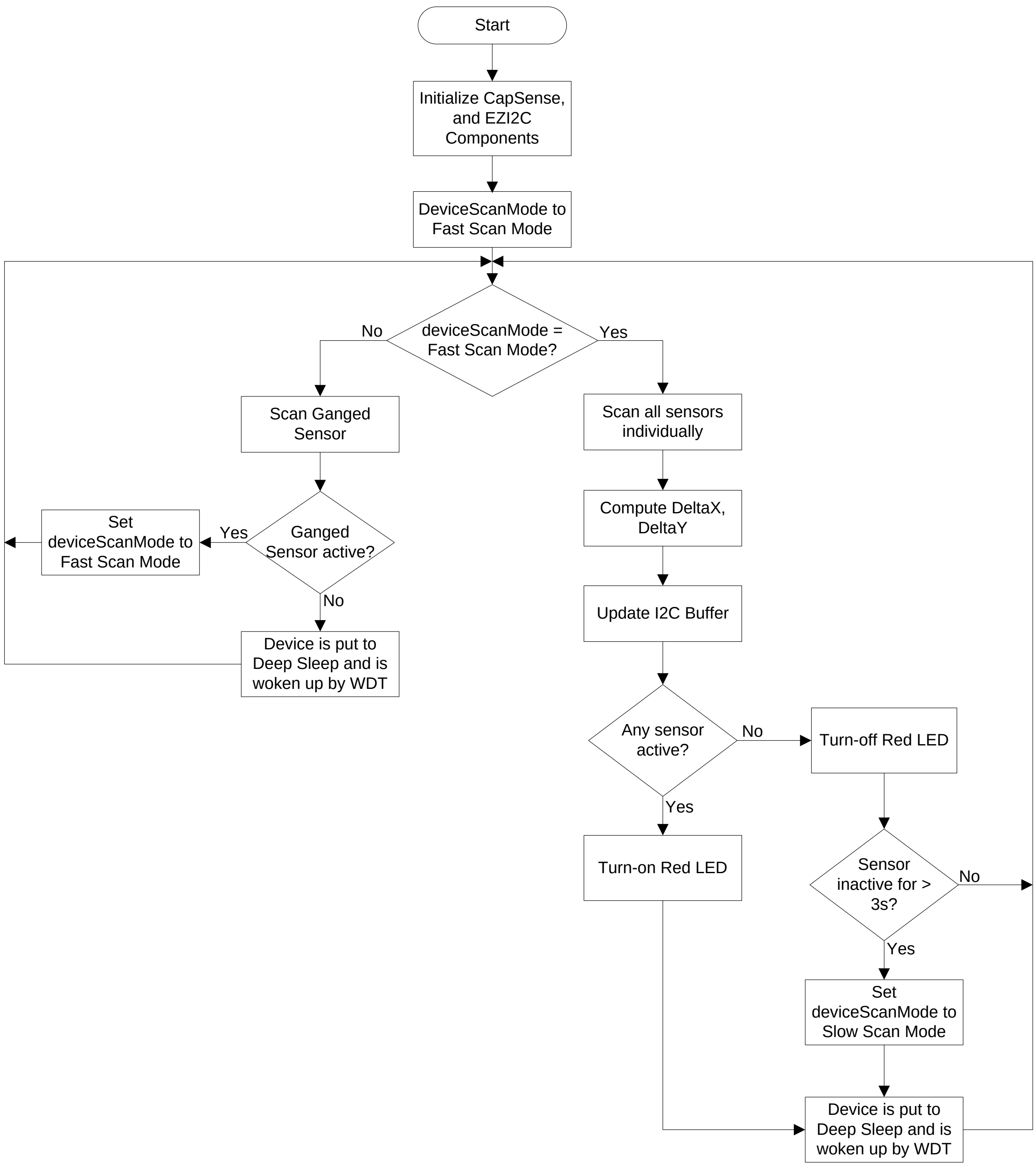


Figure 3. Firmware Flowchart



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 Device Selector, change the pin assignments in the *.cydwr* settings, and re-tune the CapSense sensors. For the tuning procedure, see AN85951 – PSoC 4 and PSoC Analog Coprocessor CapSense Design Guide.

This code example uses the internal shield drive mode and therefore the shield performance, i.e., shield switching between V_{REF} and ground depends on the operating voltage of the device. For the tuning procedure, see AN85951 – PSoC 4 and PSoC Analog Coprocessor CapSense Design Guide.

After plugging the kit to the USB port, CapSense initialization is delayed by 100 ms to ensure the baseline is initialized to a stable raw count value. This is required only when the power supply is not stable while the CapSense calibration is in process.

Note: The maximum response time to detect an approaching finger in the Slow Scan mode is equal to the refresh interval in the Slow Scan mode multiplied by the IIR filter delay. Therefore, the response to wake-on-approach appears to be slow. You can reduce the refresh interval (macro `LOOP_TIME_SLOWSCANMODE` in *main.c* file) in the Slow Scan mode to achieve a fast response time at the expense of a high average power consumption.

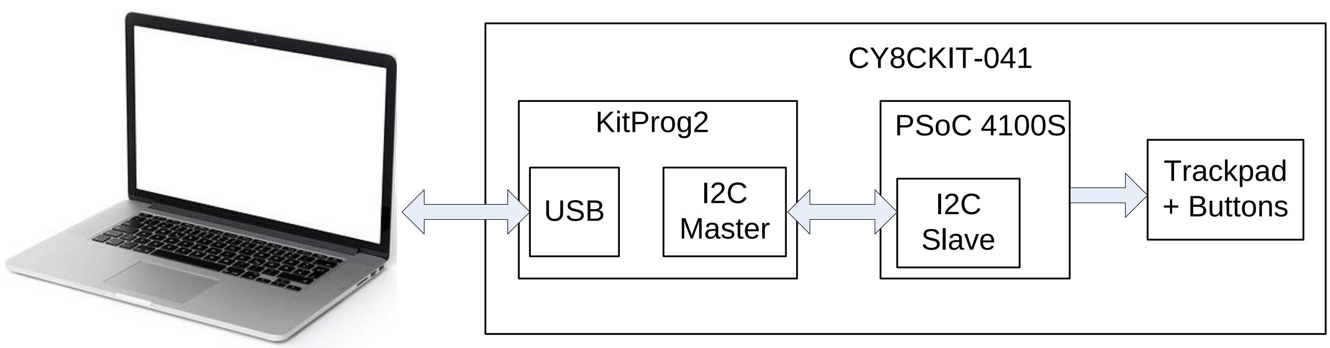
Hardware Setup

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

Software Setup

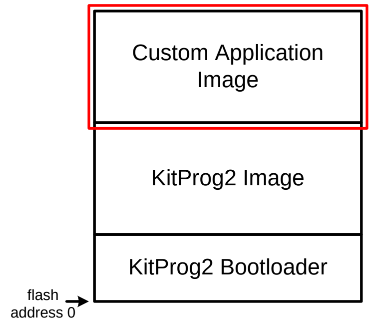
KitProg2 on the CY8CKIT-041-41XX PSoC 4100S Pioneer Kit is used to read the trackpad and button data from PSoC 4100S device via I^2C communication and send this data to PC via USB communication, as shown in Figure 4.

Figure 4. CY8CKIT-041 Interfaced as a USB-HID device to a Windows PC



To support programming/debugging and I^2C-USB HID mode, KitProg2 (a PSoC 5LP device) supports dual-image bootloading as shown in Figure 5; the KitProg2 firmware (for programming/debugging) is the first image and a custom application (for I^2C-USB HID) is loaded as a second image. In this code example, the custom application image acts as a USB-HID interface to the PC and reads the trackpad data from the PSoC 4100S device via an I^2C interface.

Figure 5. Dual-Image Bootloadable in KitProg2

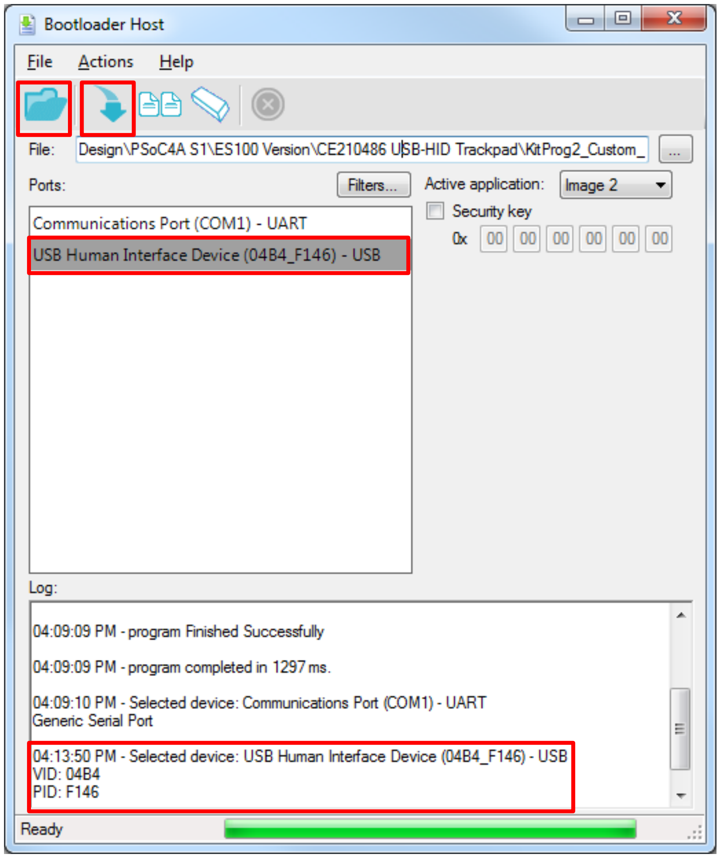


Bootloading KitProg2 with Custom HID

To bootload the custom application image to KitProg2, follow the steps below:

1. Press and hold the switch SW3 while connecting the kit to the PC. If the switch is pressed for more than 100 ms during power up, KitProg2 enters the bootloader mode. The amber status LED will start blinking to indicate that KitProg2 has entered the bootloader mode.
2. In PSoC Creator, navigate to **Tools** > **Bootloader Host**. In the bootloader host window, confirm that the USB bootloader is listed in the **Ports** window as shown in Figure 6. If it does not appear as shown, click on "**Filters**…" and verify that "**Show USB Devices**" is selected and that the VID and PID are set to **04B4** and **F146** (you can also leave the PID blank).

Figure 6. Bootloading Custom USB-HID on KitProg2



3. Click **Open File** and navigate to <install directory>\Cypress\CY8CKIT-041-41XX PSoC 4100S Pioneer Kit\1.0\Firmware\PSoC 4100S\CE216892 USB-HID Trackpad. Select the *KitProg2_Custom_2.cyacd* file and click **OK**.
4. Click **Program** and confirm that the bootloadable is programmed to the KitProg2 device by observing that the Green and Red status LEDs are turned ON. KitProg2 now interfaces to the PC as a USB HID Mouse.

   Note: It may take a few seconds for the kit to enumerate as an HID mouse on the PC.
5. To switch KitProg2 from the custom HID application to programming/debugging mode, press the switch SW3 on the kit for more than two seconds. Once you release SW3, the amber status LED should be ON. If the amber LED shows a breathing effect, press and release SW3 quickly (less than two seconds). For details on the KitProg2 operation, refer to KitProg2 User Guide.

Components

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

Table 1. List of PSoC Creator Components

Component	Instance Name	Version	Hardware Resources
CapSense	CapSense	v3.10	CSD and 18 GPIO pins
EZI2C Slave (SCB mode)	EZI2C	v3.20	SCB and 2 GPIO Pins
Digital Output Pin	Red_LED	v2.20	1 GPIO pin

Parameter Settings

CapSense

Figure 7 through Figure 12 show the CapSense Component settings that are changed from default values. See the CapSense Component datasheet for additional information.

Figure 7. CapSense Component – Basic Tab Configuration

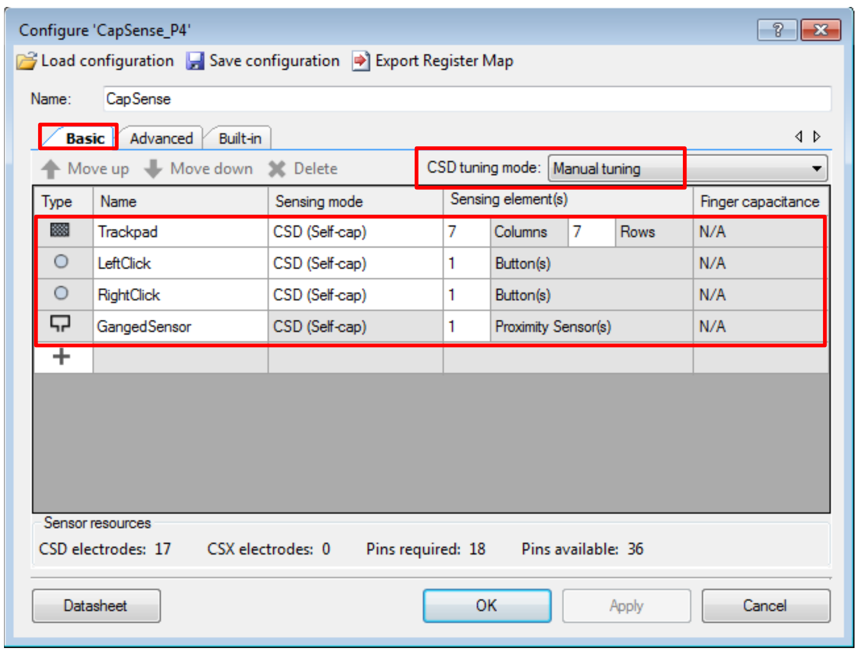


Figure 8. CapSense Component – Advanced Tab CSD Configuration

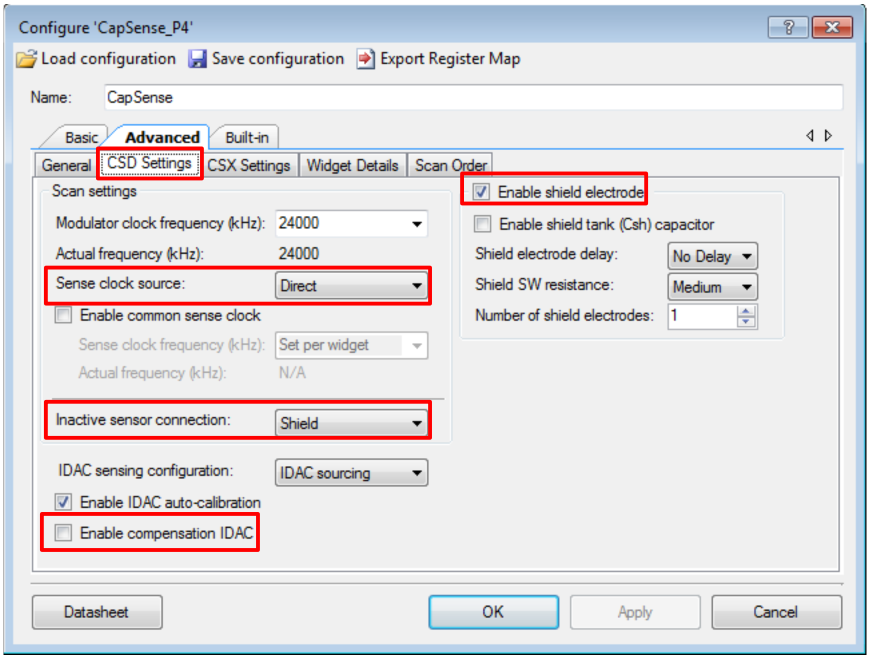


Figure 9. CapSense Component – Advanced Tab Widget Details Configuration for Trackpad

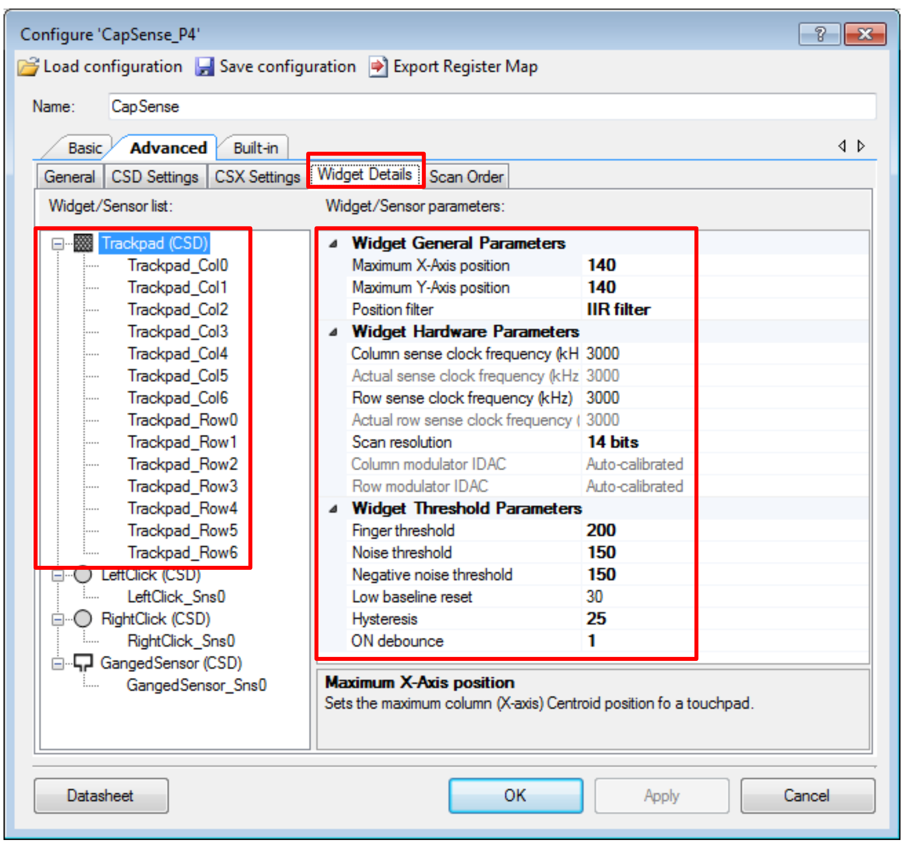


Figure 10. CapSense Component- Advanced Tab Widget Details Configuration for Left/Right Click Button

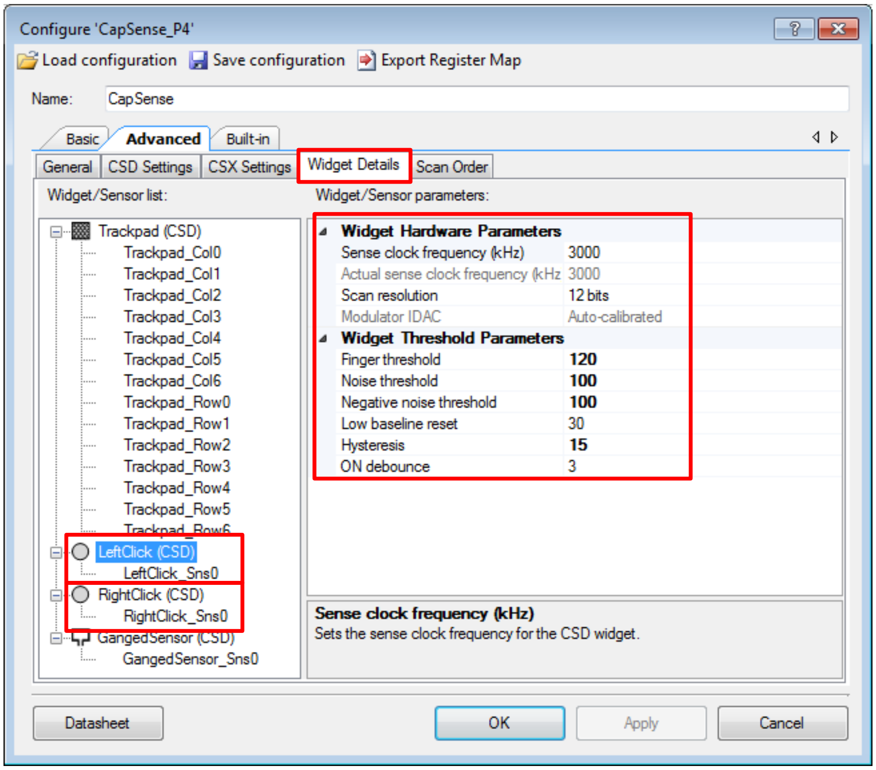


Figure 11. CapSense Component- Advanced Tab Widget Details Configuration for GangedSensor

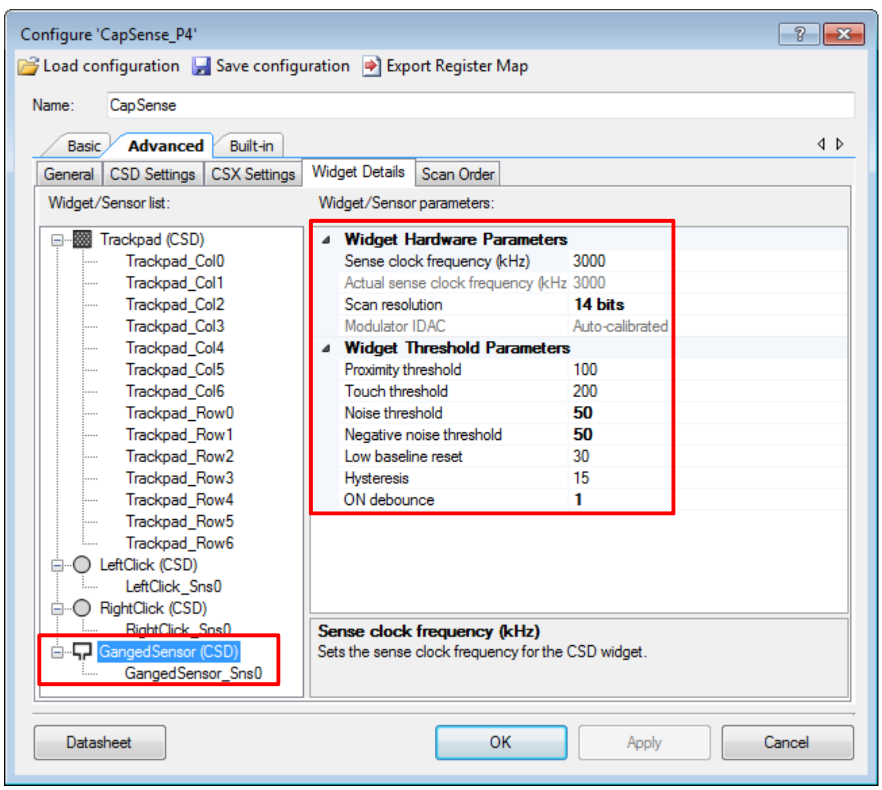
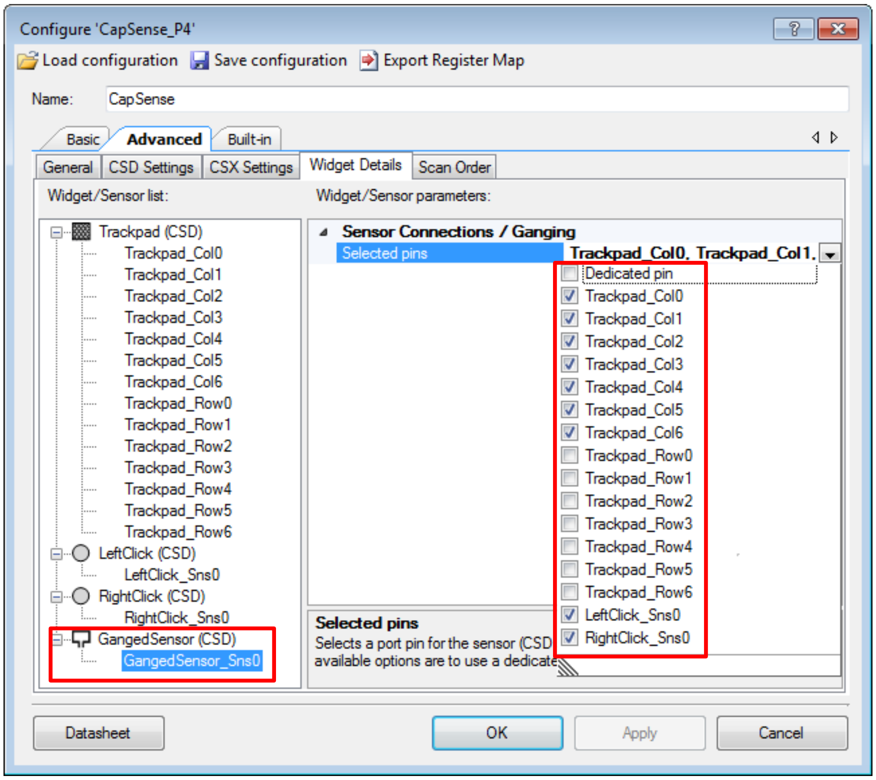


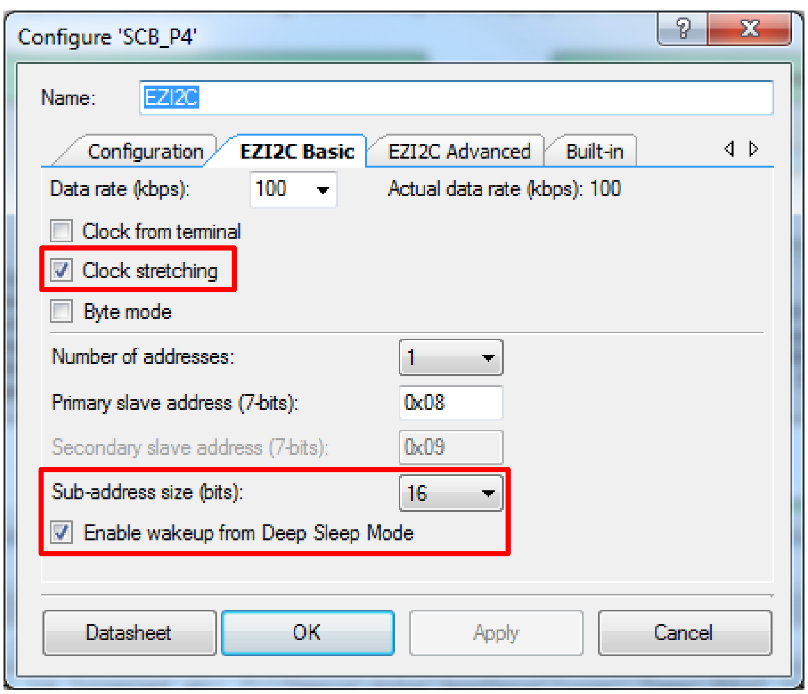
Figure 12. CapSense Component - Advanced Tab Sensor Connections for GangedSensor



EZI2C Slave

Figure 13 shows the non-default EZI2C Slave Component settings. See the SCB Component datasheet for additional information.

Figure 13. EZI2C Component Basic Tab Configuration



Design-Wide

Figure 14 and Figure 15 show the non-default **.cydwr** settings for the project.

Figure 14. cydwr Pins Tab Settings

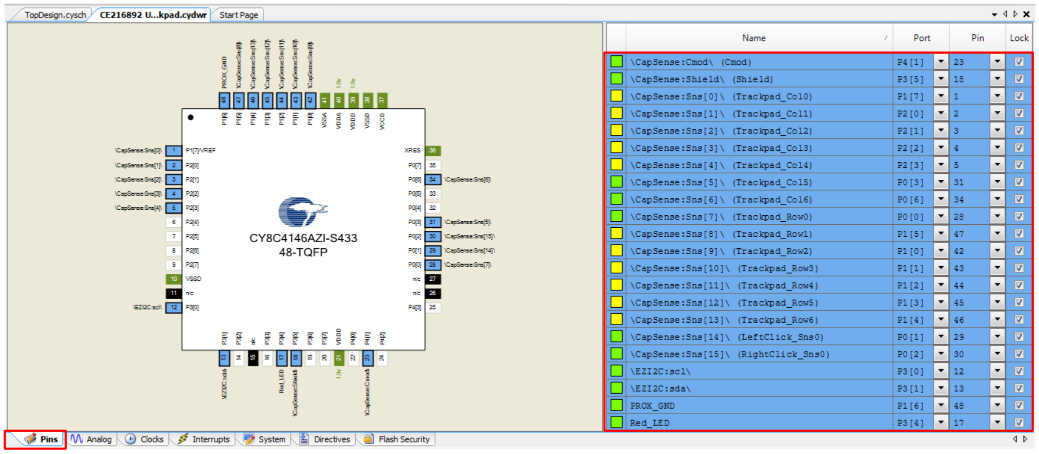
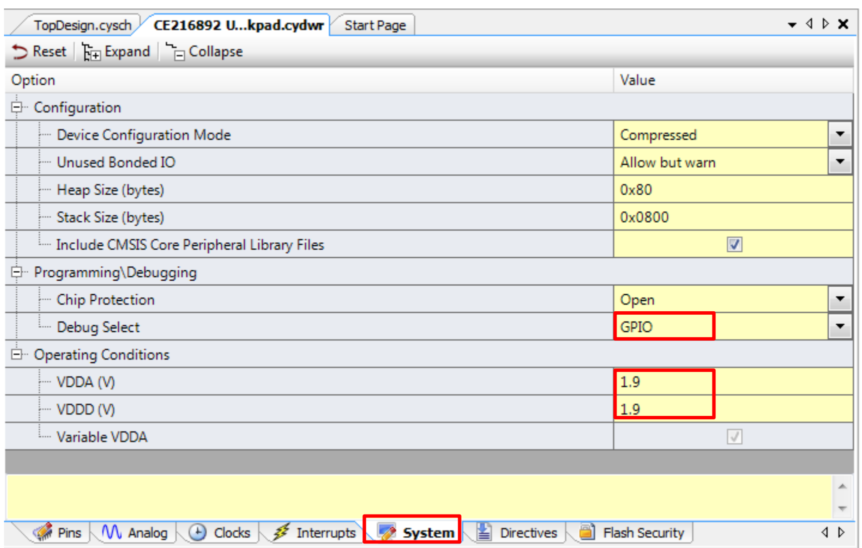


Figure 15. .cydwr System Tab Setting



Note: For PSoC 4 S-Series devices, the CapSense V_{REF} voltage is set based on the VDDA setting in the cydwr tab as per the following table.

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 cydwr, 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 sensors.

Operation

1. The custom application should be bootloaded before testing the project. See the Software Setup section for details on how to bootload the custom application.
2. Select the *CE216892 USB-HID Trackpad.cywrk* file in the PSoC Creator Start Page, under **Examples and Kits** > **Kits** > **CY8CKIT-041-41XX**. Select a location to save the code example.
3. Build the project (**Build** > **CE216892 USB-HID Trackpad**).
4. Connect the PSoC 4100S Pioneer Kit to your computer using the USB cable provided. Ensure that the kit is in programming/debugging mode (Amber status LED is always ON). If the kit is not in the programming/debugging mode, see KitProg2 User Guide for details on how to switch KitProg2 to programming/debugging mode.
5. Program the PSoC 4100S device (**Debug** > **Program**). See the CY8CKIT-041-41XX Kit Guide for details on programming the kit.
6. Press the switch SW3 on the kit for more than five seconds to switch KitProg2 from programming mode to custom application mode. The red and green status LEDs will be on when the custom application mode is running.
7. Press the RESET switch on the kit to initialize the PSoC 4100S device. Wait for the device to enumerate as an HID mouse on the PC.
8. Hover your finger over the trackpad or button sensors at a distance of 2 cm from the kit and notice that the red LED is turned ON indicating that proximity is detected.
9. Slide your finger on the trackpad to control the mouse pointer on your windows PC. Touch the CapSense buttons to trigger the mouse left-click and right-click actions.
10. Using the water dropper provided, pour small water droplets (< 3 mm) on the trackpad. Notice that the trackpad does not false trigger even when water droplets are on the trackpad. Remove the water droplets and notice that the trackpad works normally when a finger is present on the trackpad.
11. To switch KitProg2 back to programming/debugging mode, press the switch SW3 for more than two seconds and then release. Confirm that the amber LED is continuously ON, indicating that KitProg2 is re-enumerated in the KitProg2 programmer and debugger mode.

The example project supports viewing CapSense data via the CapSense Tuner. To read the CapSense data via the CapSense Tuner, the `TUNER_ENABLE` macro in the *main.c* file should be set to '1' and KitProg2 should be in the programming/debugging mode. For details on how to launch the tuner and read the CapSense data, refer to the CapSense Component Datasheet.

Note: Because the Tuner uses I^2C for reading CapSense data, the mouse functionality will be disabled when the TUNER_ENABLE macro is set to '1'.

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 all relevant application notes, code examples, knowledge base articles, device datasheets, and Component datasheets.

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
PSoC Creator Component Datasheets		
CapSense	Supports capacitive touch sensing	
EZI2C Slave	Supports I^2C slave operation	
Pins	Supports connection of hardware resources to physical pins	
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:

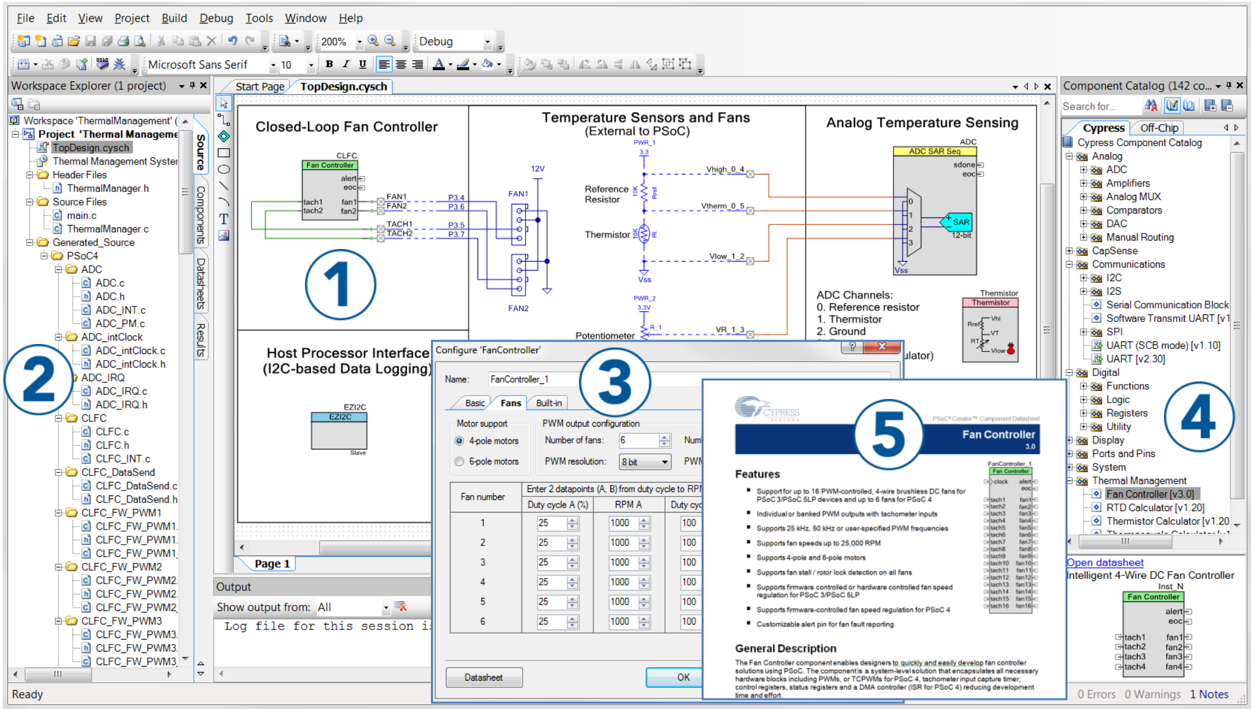
- **Overview:** PSoC Portfolio, PSoC Roadmap
- **Product Selectors:** PSoC 1, PSoC 3, PSoC 4, or PSoC 5LP. In addition, PSoC Creator includes a device selection tool.
- **Datasheets** describe and provide electrical specifications for the PSoC 3, PSoC 4, and PSoC 5LP device families.
- **CapSense Design 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 easy-to-use and inexpensive development platform. This kit include 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 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 16 – 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 16. PSoC Creator Features



Document History

Document Title: CE216892 - USB HID Trackpad

Document Number: 002-16892

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5530816	SRDS	11/23/2016	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