CE210291 - PSoC4 CapSense® Low Power One Button

Objective

This code example demonstrates how to implement a low-power CapSense button with an average current consumption of 6uA per button.

Overview

This code example implements a CapSense button. Using the low power modes available in the PSoC device, an average current of 6 uA is achieved for a button sensor when no touch is present. The example supports CapSense tuner to read the CapSense debug data such as raw count, baseline and difference count.

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, PSoC 4 M-Series, PSoC 4 L-Series, PSoC 4 S-Series, PSoC 4 BLE and PRoC BLE devices

Related Hardware: CY8CKIT-041-40XX, CY8CKIT-041-41XX, CY8CKIT-042, CY8CKIT-044, CY8CKIT-046, and CY8CKIT-042-BLE.

Design

The project demonstrates the core functionality of the CapSense component. This code example uses CapSense, EZI2C Slave and Global Signal Reference components. Figure 1 shows the component schematics of 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 and button touch position information to the external host/tuner via I^2C.

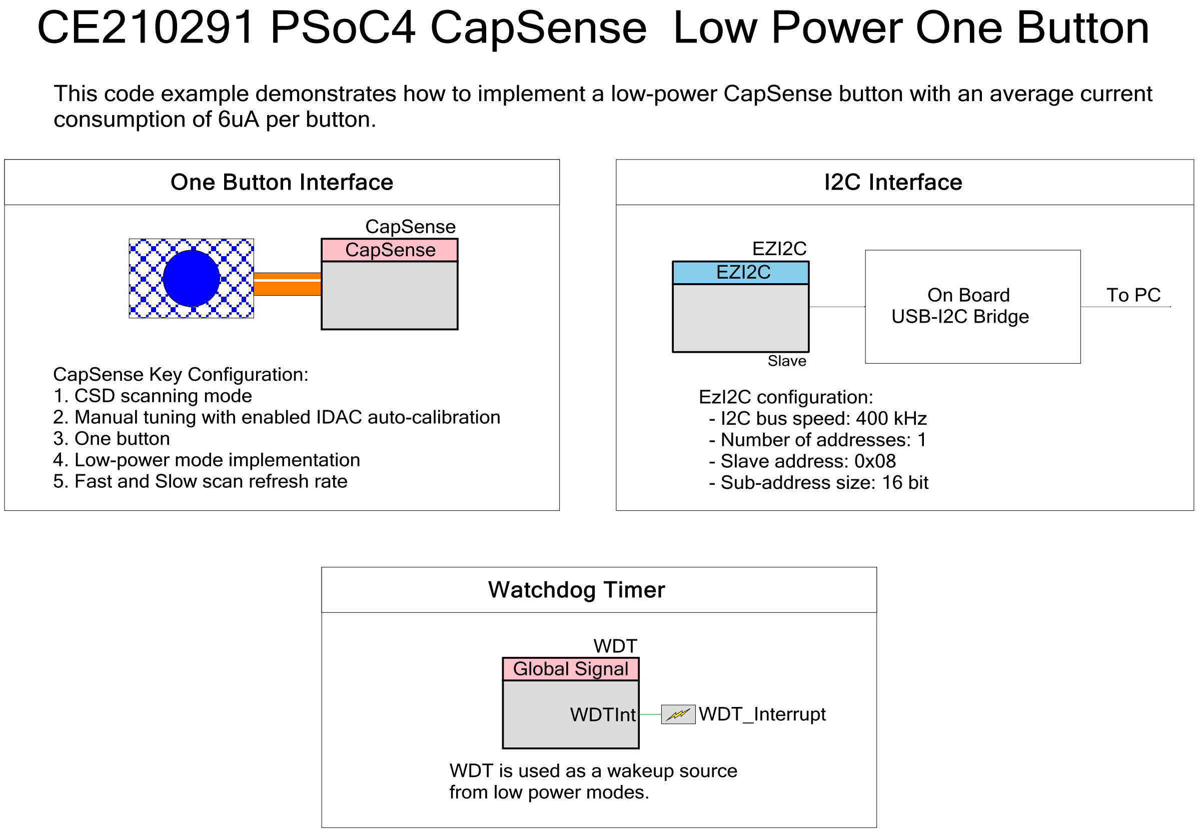
To reduce the power consumed by the PSoC device and provide optimum touch response, this code example implements two modes: Fast Scan Mode and Slow Scan Mode. When the user is interacting with the buttons, the PSoC device is in Fast Scan Mode and when the user is not interacting with the buttons for a specific duration, the Slow Scan Mode is used.

In the Fast Scan Mode, the button sensor is scanned at a refresh rate of 33 Hz (or scan interval of 30 ms) and the RGB LED is driven based on the button status. The PSoC device is put to Deep Sleep after the CapSense data is processed. The The Global Signal Reference component which is configured as a Watchdog timer (WDT) timer is used to periodically wake up the device from Sleep mode. This mode provides optimum touch response, but consumes higher average current (< 30uA) when compared to Slow Scan Mode.

The Slow Scan Mode is similar to the Fast Scan Mode except that the refresh rate is 5 Hz (or scan interval is 200 ms). The Slow Scan Mode consumes a lower average current of 5 uA per button, but with a slower touch response. Once touch is detected in Slow Scan Mode, the PSoC device switches to Fast Scan Mode to provide optimum touch response at the expense of higher current consumption.

The firmware flowchart is shown in Figure 2.

Figure 1. Component Schematics



CE210291 PSoC4 CapSense Low Power One Button

This code example demonstrates how to implement a low-power CapSense button with an average current consumption of 6uA per button.

One Button Interface

CapSense Key Configuration:
1. CSD scanning mode
2. Manual tuning with enabled IDAC auto-calibration
3. One button
4. Low-power mode implementation
5. Fast and Slow scan refresh rate

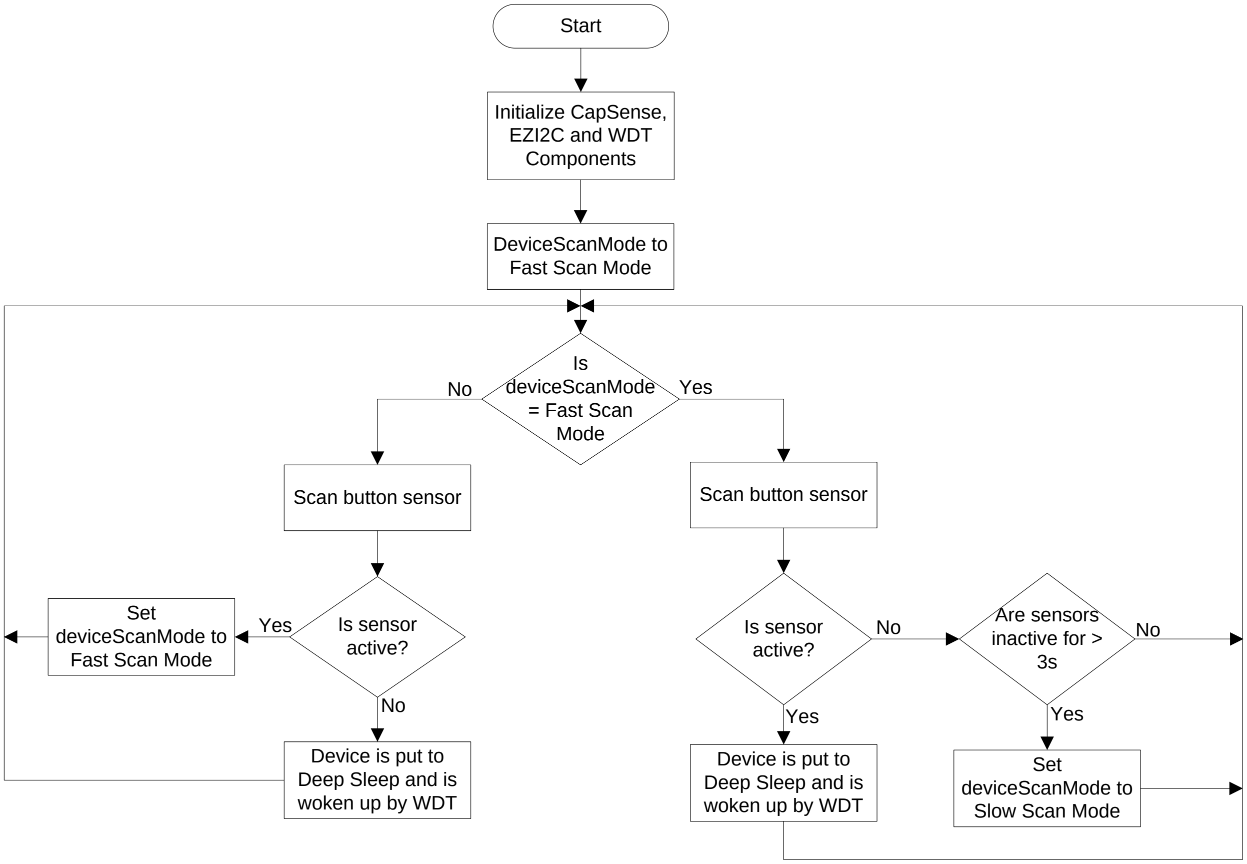
I2C Interface

EzI2C configuration:
- I2C bus speed: 400 kHz
- Number of addresses: 1
- Slave address: 0x08
- Sub-address size: 16 bit

Watchdog Timer

WDT is used as a wakeup source from low power modes.

Figure 2. Firmware Flowchart


Start
Initialize CapSense, EZI2C and WDT Components
DeviceScanMode to Fast Scan Mode
Is deviceScanMode = Fast Scan Mode
No
Yes
Scan button sensor
Scan button sensor
Is sensor active?
Yes
No
Set deviceScanMode to Fast Scan Mode
Device is put to Deep Sleep and is woken up by WDT
Is sensor active?
No
Yes
Are sensors inactive for > 3s
No
Yes
Device is put to Deep Sleep and is woken up by WDT
Set deviceScanMode to Slow Scan Mode

Design Considerations

The code example is designed for the PSoC 4000S 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 the steps below:

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.

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

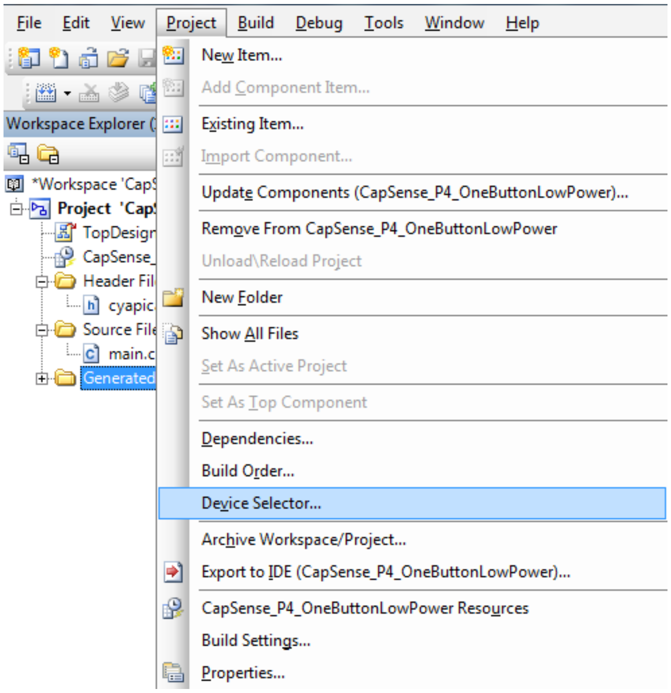


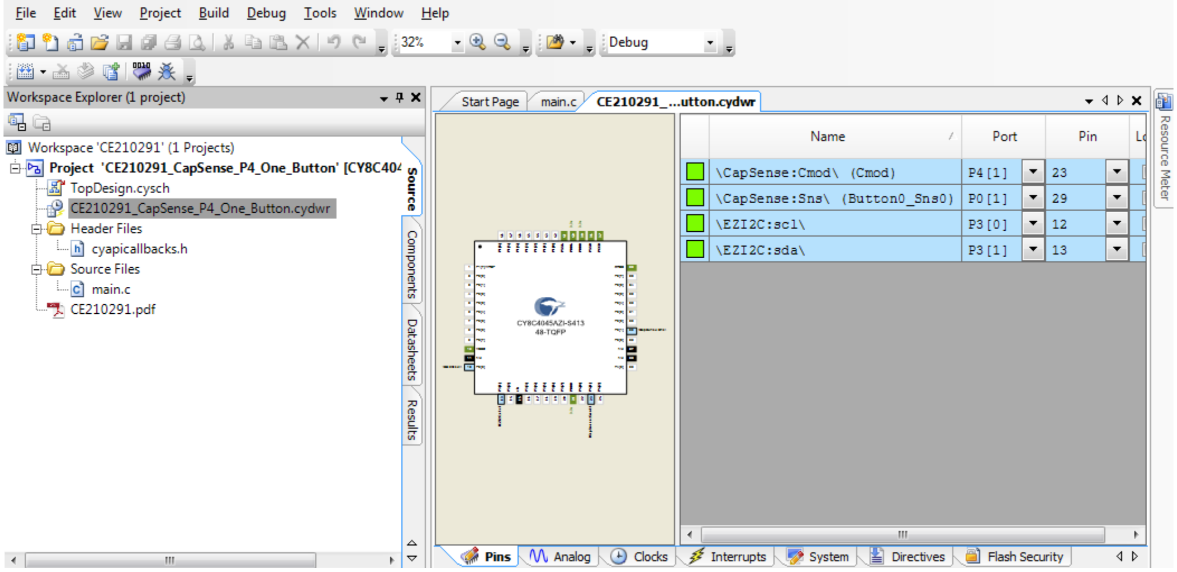
Table 1. Pin Assignment for the PSoC4 CapSense Low Power One Button Project

Pin name	Development kits					
	CY8CKIT-041-40XX (CY8C4045AZI-S423)	CY8CKIT-041-41XX (CY8C4146AZI-S433)	CY8CKIT-042 (CY8C4245AXI-483)	CY8CKIT-042-BLE (CY8C4247LQI-BL483)	CY8CKIT-044 (CY8C4247AZI-M485)	CY8CKIT-046 (CY8C4248BZI-L489)
\Capsense: Cmod\	P4[1]	P4[1]	P4[2]	P4[0]	P4[2]	P4[2]
\CapSense: Sns[0]\ (Button0_Sns0)	P0[1]	P0[1]	P1[1]	P2[1]	P4[4]	P0[6]
\EZI2C: scl\	P3[0]	P3[0]	P3[0]	P3[5]	P4[0]	P4[0]
\EZI2C: sda\	P3[1]	P3[1]	P3[1]	P3[4]	P4[1]	P4[1]

2. When you select the device, pins are assigned automatically in the design wide resource file according to the device selected.

   Note: If the assigned pins are not as shown in Table 1 or you want to overwrite the existing pin assignments, double click the project’s design wide resource file (Figure 4) in the workspace explorer window and assign the pins. Refer device datasheet to use other pins.

Figure 4. Pin Assignments in Project’s Design Wide Resource File



	Name	Port	Pin
🟩	\CapSense:Cmod\ (Cmod)	P4 [1]	23
🟩	\CapSense:Sns\ (Button0_Sns0)	P0 [1]	29
🟩	\EZI2C:scl\	P3 [0]	12
🟩	\EZI2C:sda\	P3 [1]	13

3. Select 3.3 V as supply by using switch SW5. For other kits refer to the kit user guide.
4. Build the project and ensure that there are no errors or warnings.

   Note: For PSoC 4 device, before building the project, the modulator clock frequency should to be set to 12 MHz instead of 24 MHz because the maximum frequency supported is 12 MHz.

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

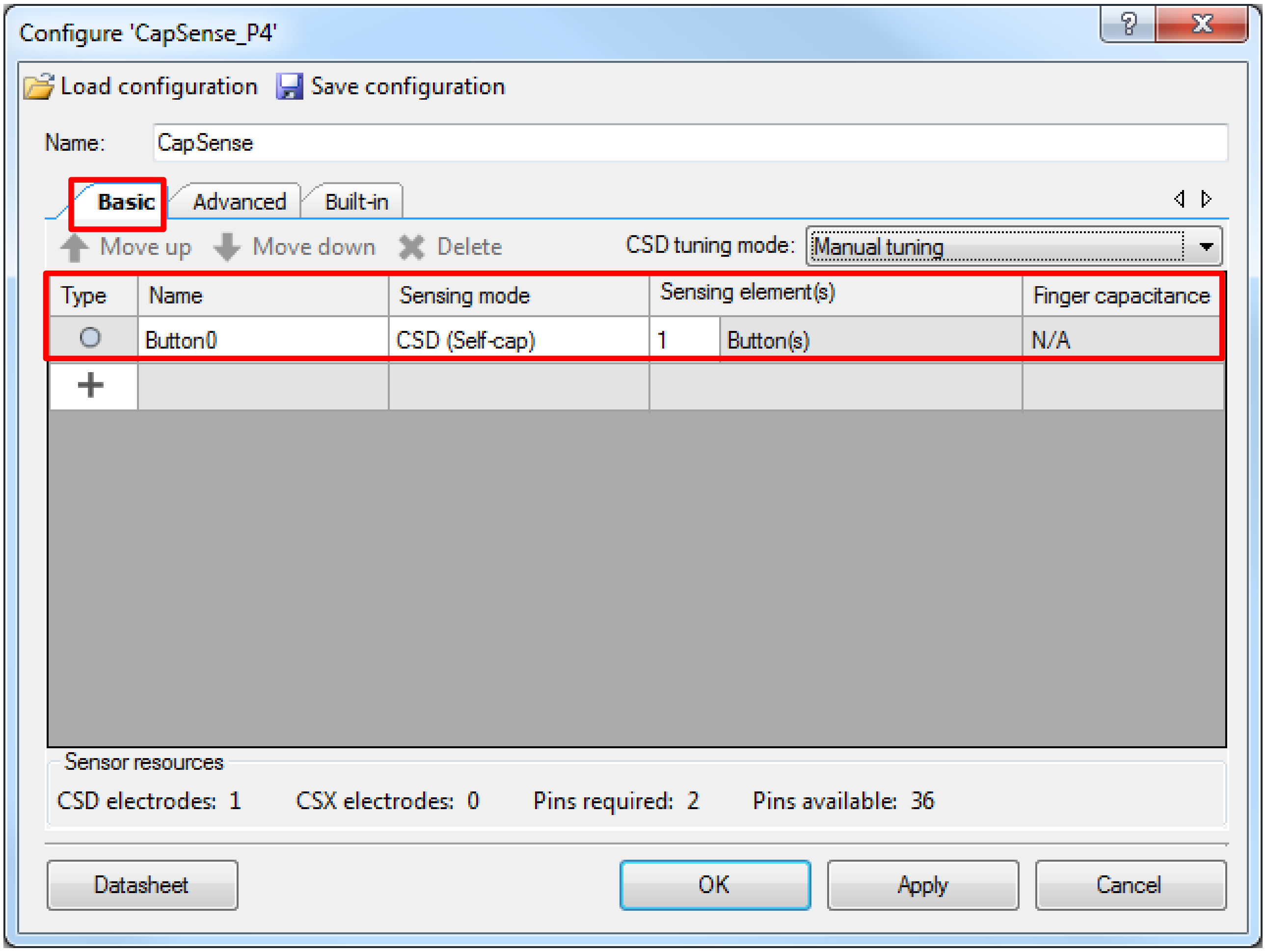
Component	Instance Name	Component Version	Hardware Resources
CapSense	CapSense	v3.0	CSD, and 2 GPIO pins
EZI2C Save (SCB mode)	EZI2C	v3.20	SCB, 2 GPIO pins
Interrupt	WDT_Interrupt	V1.70	Interrupt

Parameter Settings

CapSense

Figure 5, Figure 6 and Figure 7 show the settings for CapSense component. See the CapSense component datasheet for additional information.

Figure 5. CapSense component's Basic tab



Name: CapSense

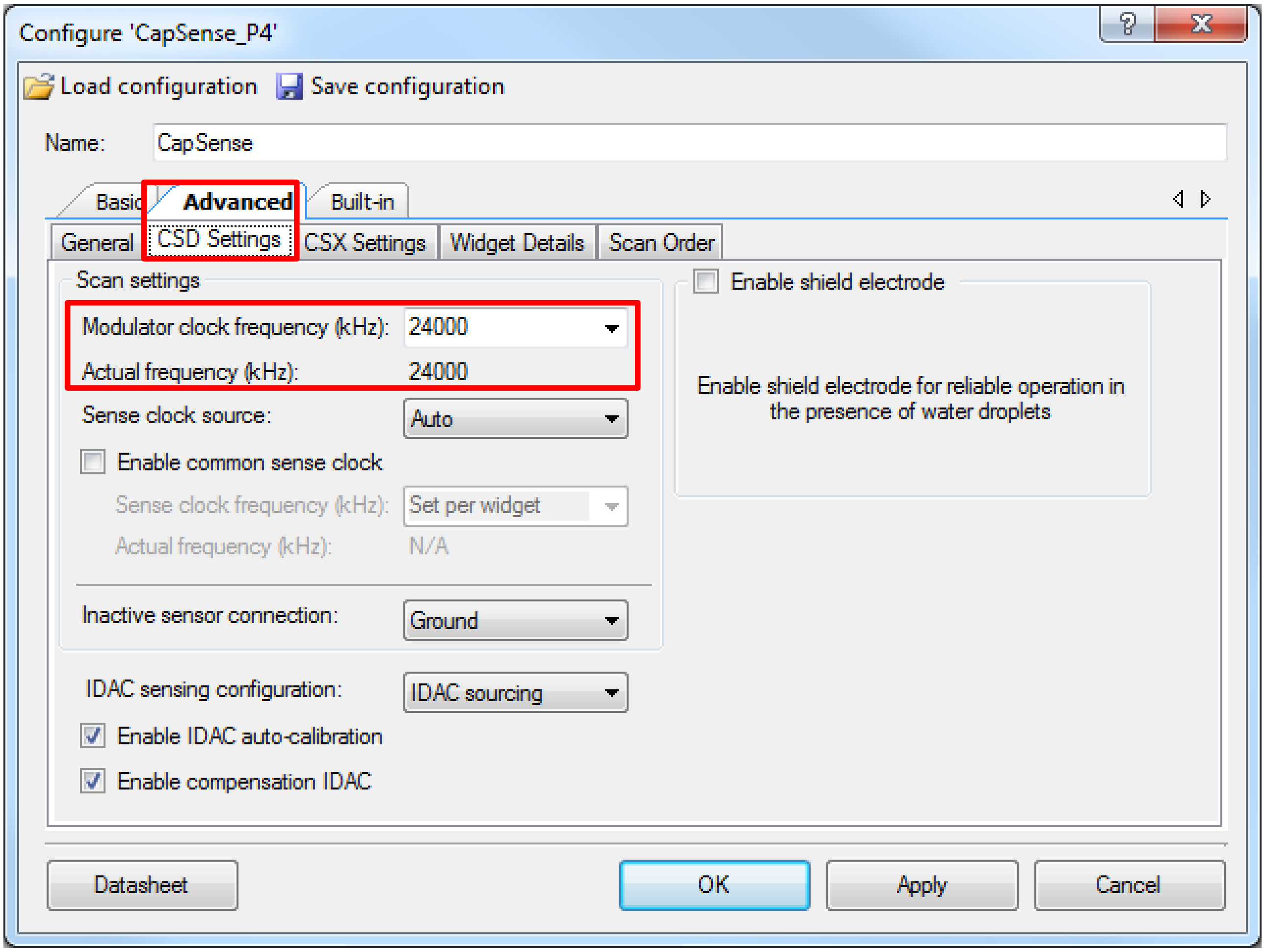
CSD tuning mode: Manual tuning

Type	Name	Sensing mode	Sensing element(s)		Finger capacitance
○	Button0	CSD (Self-cap)	1	Button(s)	N/A

Sensor resources

CSD electrodes: 1 CSX electrodes: 0 Pins required: 2 Pins available: 36

Figure 6. CapSense Component Customizer Advanced Tab (CSD Settings)



Name: CapSense

Scan settings

Modulator clock frequency (kHz): 24000

Actual frequency (kHz): 24000

Sense clock source: Auto

☐ Enable common sense clock

Sense clock frequency (kHz): Set per widget

Actual frequency (kHz): N/A

Inactive sensor connection: Ground

IDAC sensing configuration: IDAC sourcing

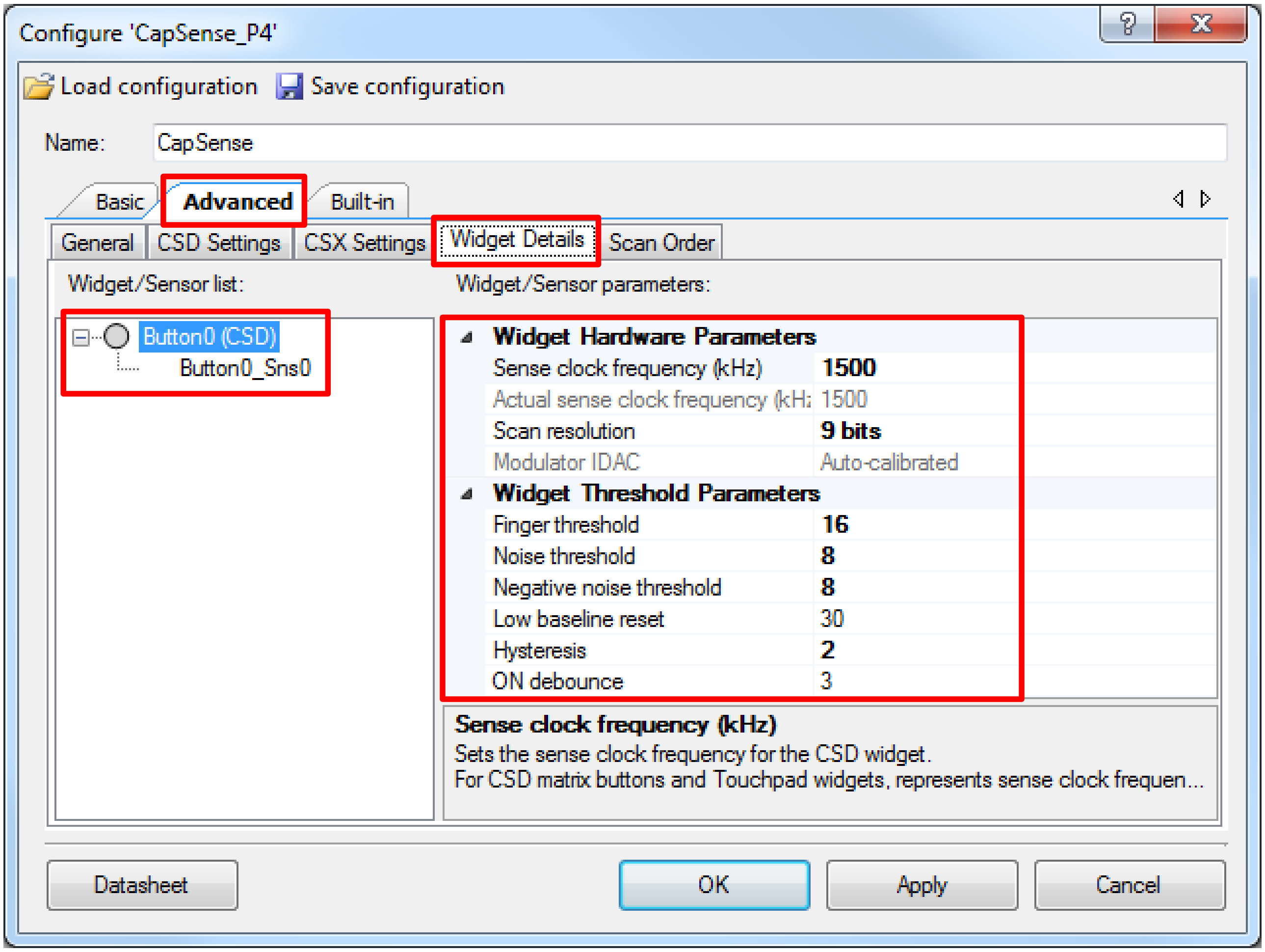
☑ Enable IDAC auto-calibration

☑ Enable compensation IDAC

☐ Enable shield electrode

Enable shield electrode for reliable operation in the presence of water droplets

Figure 7. CapSense Component Widget Details Settings



Configure 'CapSense_P4'

Load configuration Save configuration

Name: CapSense

Widget/Sensor list:

- Button0 (CSD)
  - Button0_Sns0

Widget/Sensor parameters:

Widget Hardware Parameters	
Sense clock frequency (kHz)	**1500**
Actual sense clock frequency (kHz	1500
Scan resolution	**9 bits**
Modulator IDAC	Auto-calibrated
Widget Threshold Parameters	
Finger threshold	**16**
Noise threshold	**8**
Negative noise threshold	**8**
Low baseline reset	30
Hysteresis	**2**
ON debounce	3

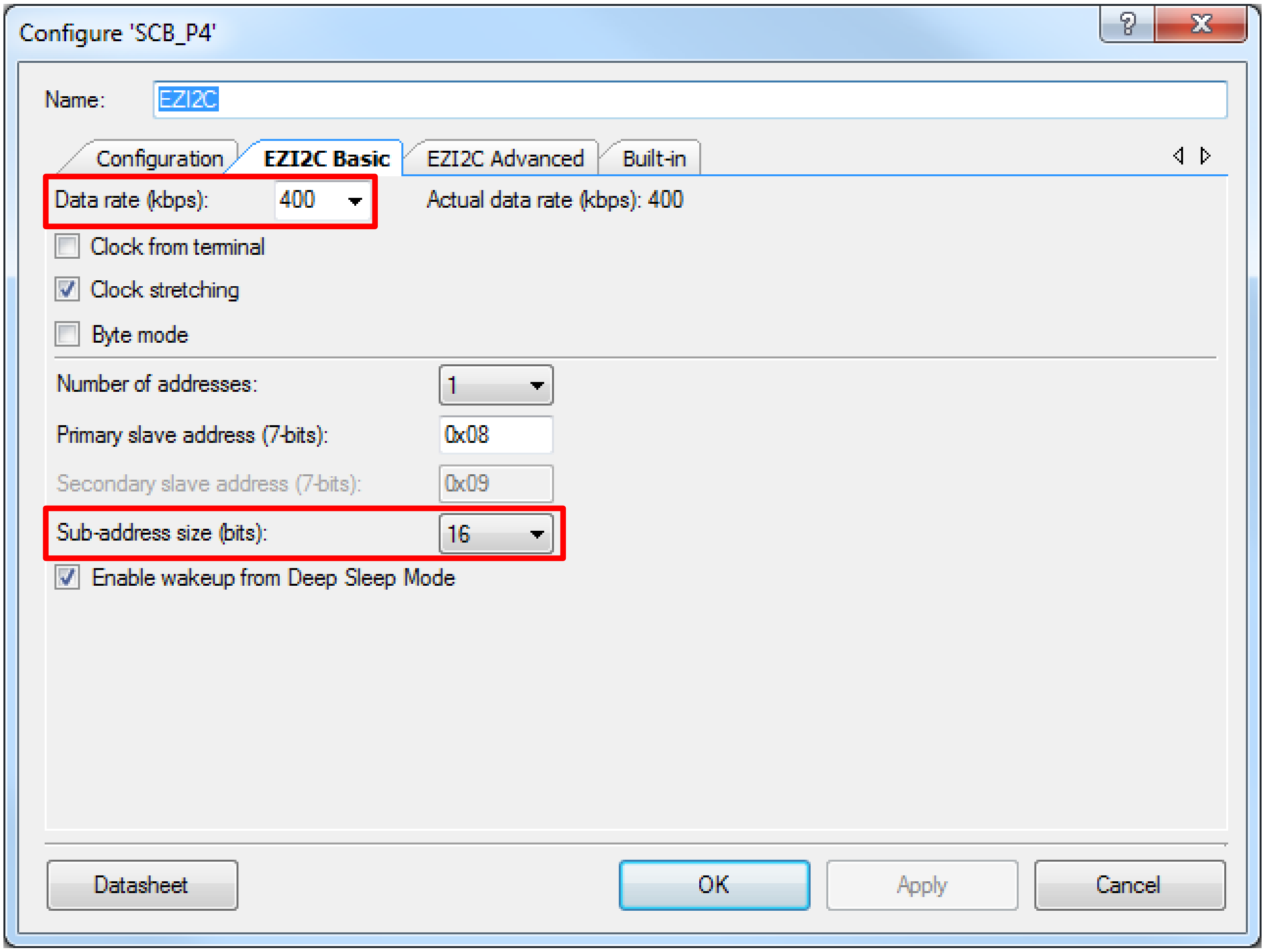
Sense clock frequency (kHz)

Sets the sense clock frequency for the CSD widget.
For CSD matrix buttons and Touchpad widgets, represents sense clock frequen...

EZI2C Slave Component

Figure 8 shows the settings for EZI2C Slave Component. See the SCB component datasheet for additional information.

Figure 8. EZI2C Component’s Basic Tab



Name: EZI2C

Data rate (kbps): 400

Actual data rate (kbps): 400

☐ Clock from terminal

☑ Clock stretching

☐ Byte mode

Number of addresses: 1

Primary slave address (7-bits): 0x08

Secondary slave address (7-bits): 0x09

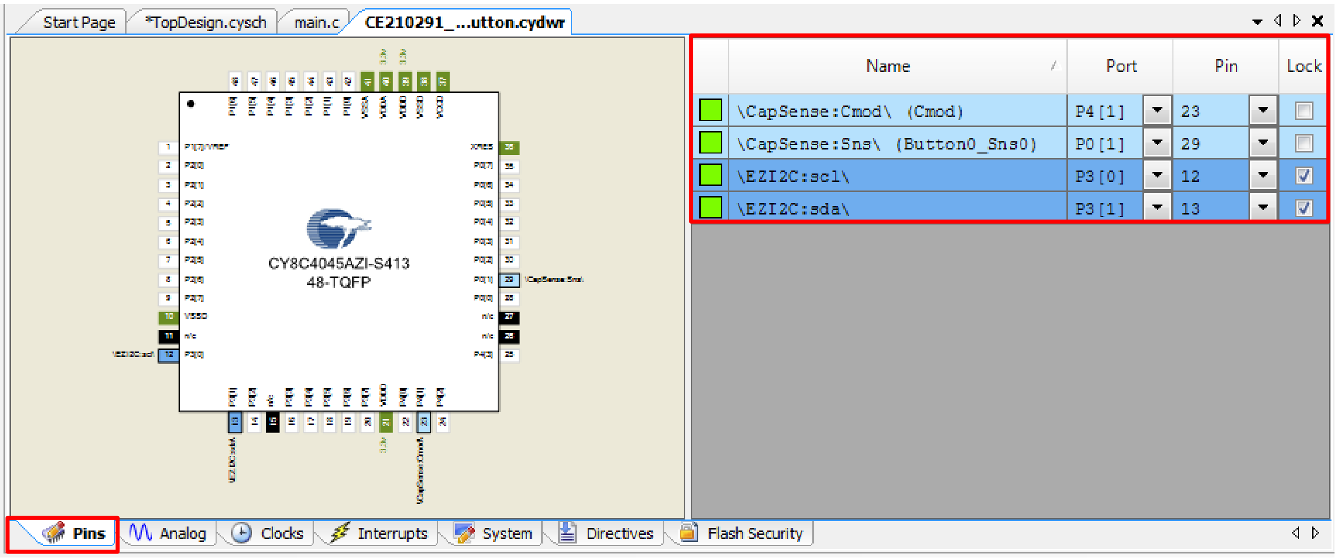
Sub-address size (bits): 16

☑ Enable wakeup from Deep Sleep Mode

Design-Wide

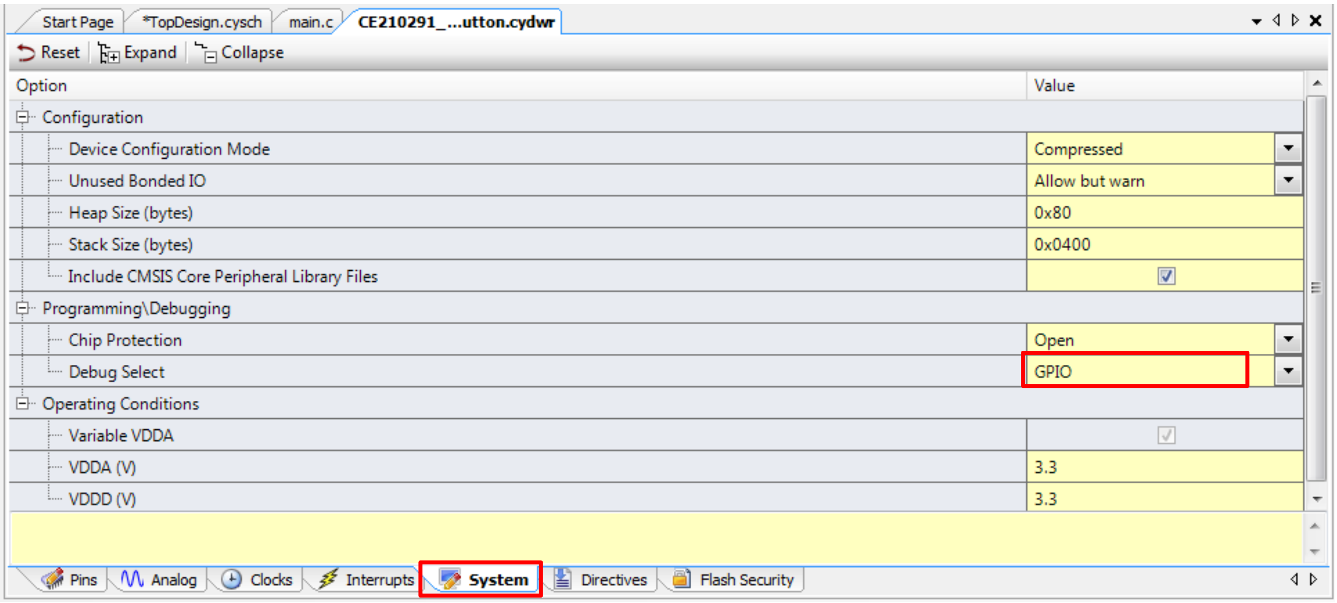
Figure 9 and Figure 10 show the non-default cydwr settings for the project.

Figure 9. CYWDR Pins Tab Settings



	Name	Port	Pin	Lock
🟩	\CapSense:Cmod\ (Cmod)	P4[1]	23	☐
🟩	\CapSense:Sns\ (Button0_Sns0)	P0[1]	29	☐
🟩	\EZI2C:scl\	P3[0]	12	☑
🟩	\EZI2C:sda\	P3[1]	13	☑

Figure 10. CYDWR System Tab Settings



Option	Value
Configuration	
Device Configuration Mode	Compressed
Unused Bonded IO	Allow but warn
Heap Size (bytes)	0x80
Stack Size (bytes)	0x0400
Include CMSIS Core Peripheral Library Files	☑
Programming\Debugging	
Chip Protection	Open
Debug Select	GPIO
Operating Conditions	
Variable VDDA	☑
VDDA (V)	3.3
VDDD (V)	3.3

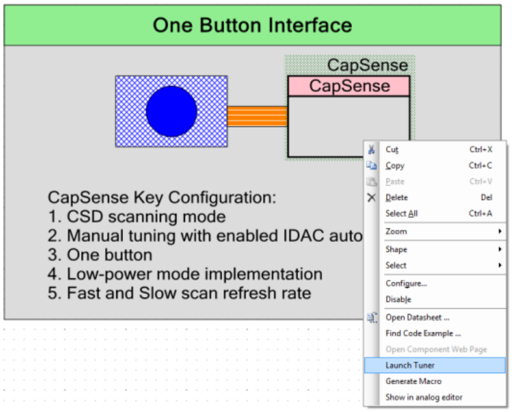
Operation

Follow the steps below to test the project on CY8CKIT-041:

1. Plug the CY8CKIT-041-40XX board into your computer's USB port and program the project into the kit. Refer to the kit user guide for details on steps for programming.
2. Connect an ammeter between test points P4.VDD and VDD on the CY8CKIT-041-40XX board. Refer to the kit user guide "Current Measurement Switch" section for complete details on power measurement steps.
3. Touch the left button on the kit and confirm that the current displayed on the ammeter in Fast Scan Mode is ~30 uA.

4. Release the finger from the button and wait for 3 seconds. Note that the current displayed on the ammeter in Slow Scan Mode is ~6 uA.
5. By default, the parameters updated in tuner is not written to the firmware. To enable this, set the macro TUNER_UPDATE_ENABLE to '1' in the main.c file. Then build the project and program the device. Note that additional current will be consumed if the conditional code is enabled.
6. To launch the Tuner GUI, right click the CapSense component, as shown in Figure 11. Then select Launch Tuner in the menu.

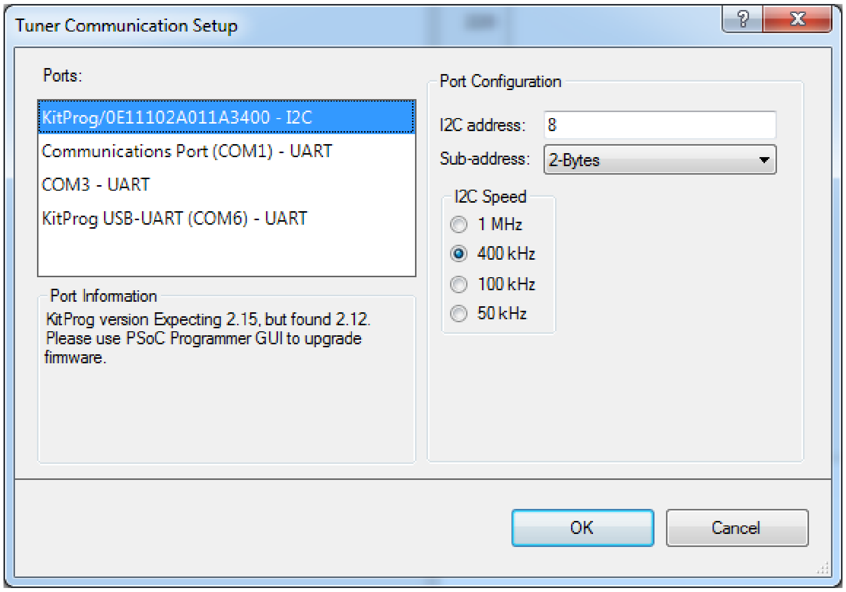
Figure 11. Launch Tuner GUI



CapSense Key Configuration:
1. CSD scanning mode
2. Manual tuning with enabled IDAC auto
3. One button
4. Low-power mode implementation
5. Fast and Slow scan refresh rate

7. Navigate to Tools/Protocol Settings in the Tuner GUI menu to set up the I2C communication (Figure 12).
   - Select the I2C port of the kit in the Ports menu.
   - Choose the I2C address, sub-address size, and speed as shown in Figure 12.

Figure 12. Setting up I2C Communication



KitProg/0E11102A011A3400 - I2C
Communications Port (COM1) - UART
COM3 - UART
KitProg USB-UART (COM6) - UART

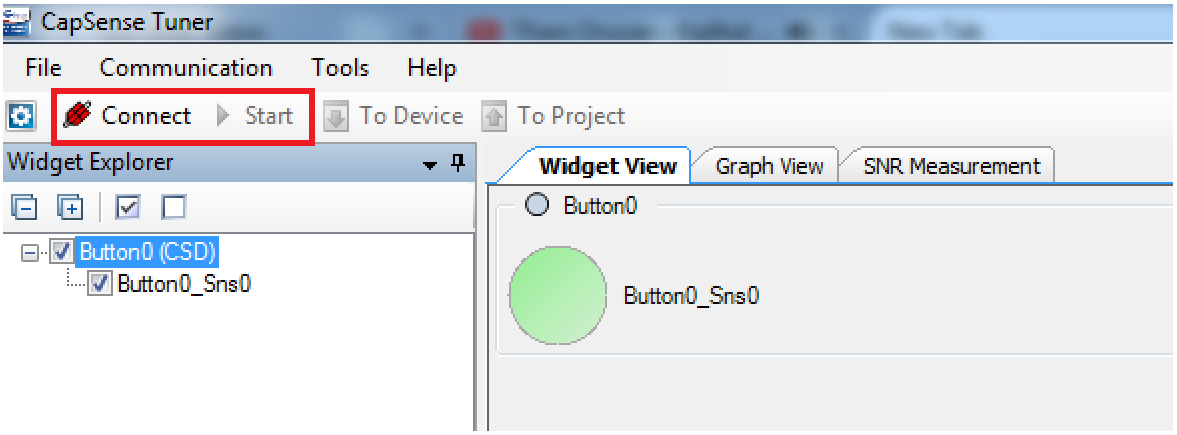
Port Information
KitProg version Expecting 2.15, but found 2.12. Please use PSoC Programmer GUI to upgrade firmware.

I2C address: 8
Sub-address: 2-Bytes

○ 1 MHz
● 400 kHz
○ 100 kHz
○ 50 kHz

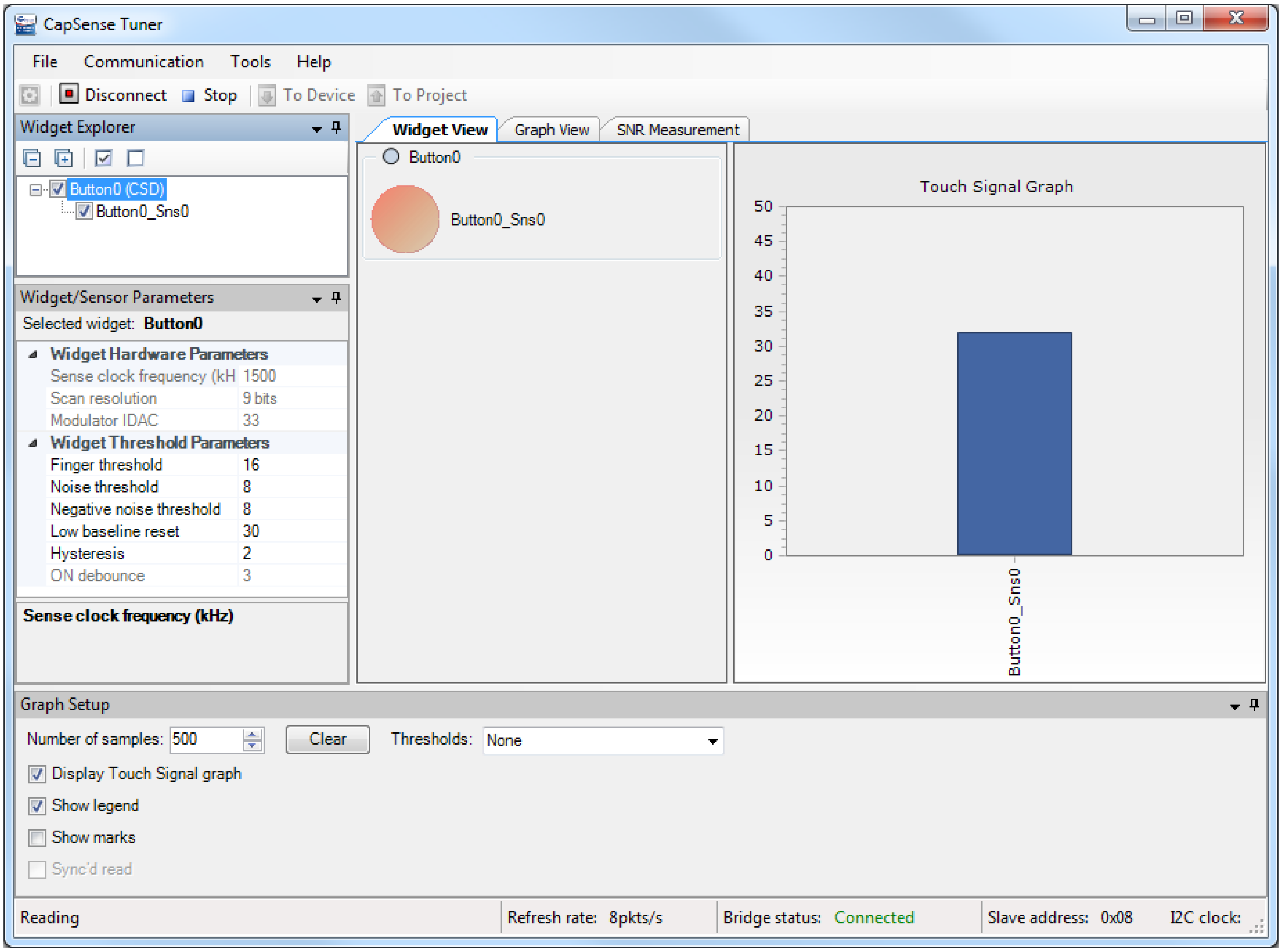
8. In the tuner GUI, click the Connect button followed by Start button (Figure 13).

Figure 13. Starting Communication



9. After establishing the I2C communication between the device and Tuner GUI, you can observe the sensor data and the button touch status.
10. Figure 14 and Figure 15 show sensor debug data such as difference count, raw count and baseline.

Figure 14. Tuner: Widget View



☑ Button0 (CSD)
  ☑ Button0_Sns0

Widget/Sensor Parameters

Selected widget: **Button0**

Widget Hardware Parameters	
Sense clock frequency (kH	1500
Scan resolution	9 bits
Modulator IDAC	33
Widget Threshold Parameters	
Finger threshold	16
Noise threshold	8
Negative noise threshold	8
Low baseline reset	30
Hysteresis	2
ON debounce	3

Sense clock frequency (kHz)

Graph Setup

Number of samples: 500

Thresholds: None

☑ Display Touch Signal graph

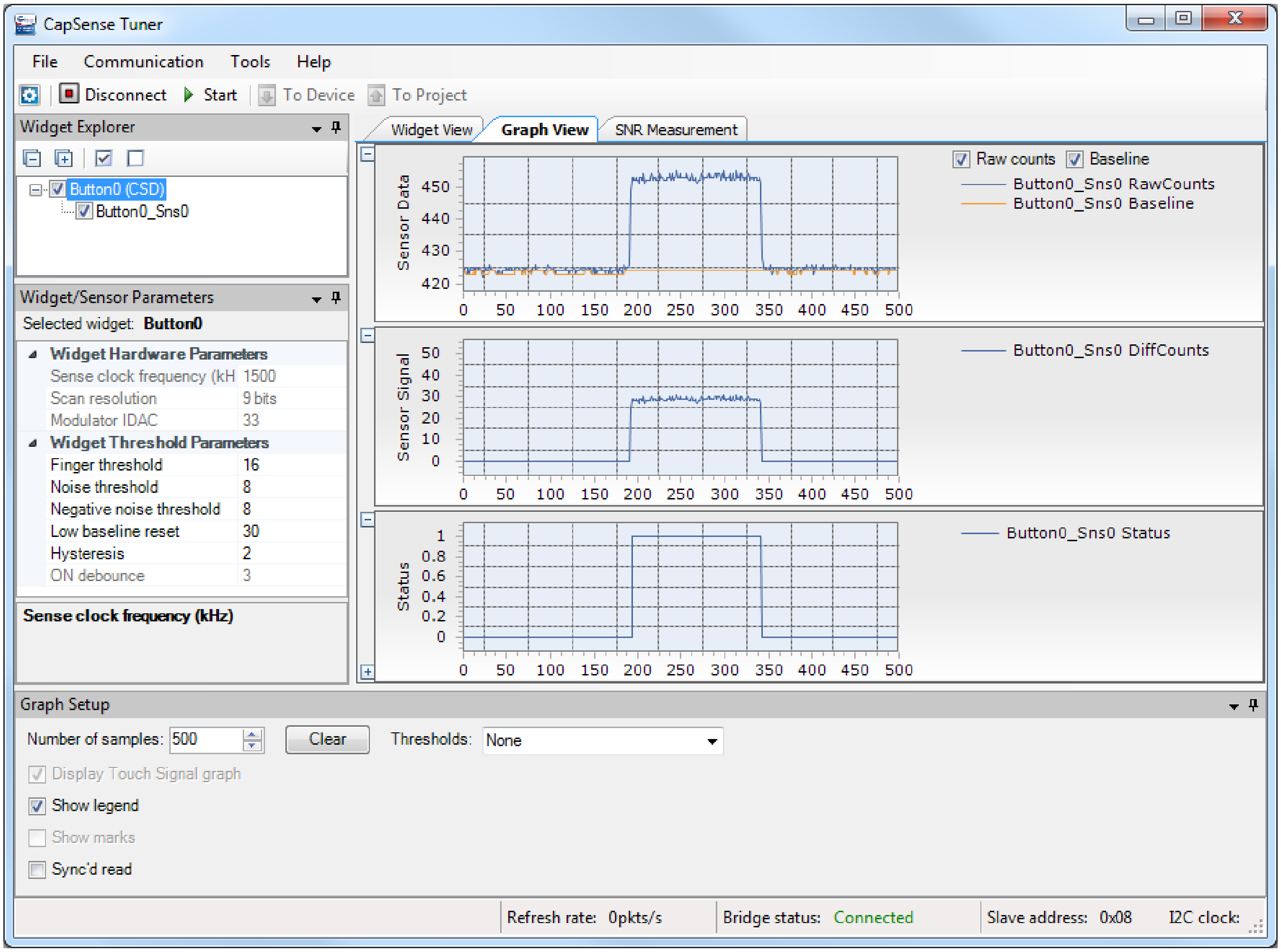
☑ Show legend

☐ Show marks

☐ Sync'd read

Reading | Refresh rate: 8pkts/s | Bridge status: Connected | Slave address: 0x08 | I2C clock:

Figure 15. Tuner: Graph View



Widget/Sensor Parameters

Selected widget: **Button0**

Widget Hardware Parameters	
Sense clock frequency (kH	1500
Scan resolution	9 bits
Modulator IDAC	33
Widget Threshold Parameters	
Finger threshold	16
Noise threshold	8
Negative noise threshold	8
Low baseline reset	30
Hysteresis	2
ON debounce	3

Sense clock frequency (kHz)

Number of samples: 500

Thresholds: None

☑ Display Touch Signal graph

☑ Show legend

☐ Show marks

☐ Sync'd read

Refresh rate: 0pkts/s

Bridge status: Connected

Slave address: 0x08

I2C clock:

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	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		
CE210290	PSoC 4 CapSense Low-Power Ganged Sensor	
PSoC Creator Component Datasheets		
CapSense	Supports capacitive touch sensing	
EZI2C Slave	Supports I2C slave operation	
Global Signal Reference	Supports Watchdog timer	
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		
CY8CKIT-042 PSoC® 4 Pioneer Kit		
CY8CKIT-042-BLE Bluetooth® Low Energy (BLE) Pioneer Kit		
CY8CKIT-044 PSoC® 4 M-Series Pioneer Kit		
CY8CKIT-046 PSoC® 4 L-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:

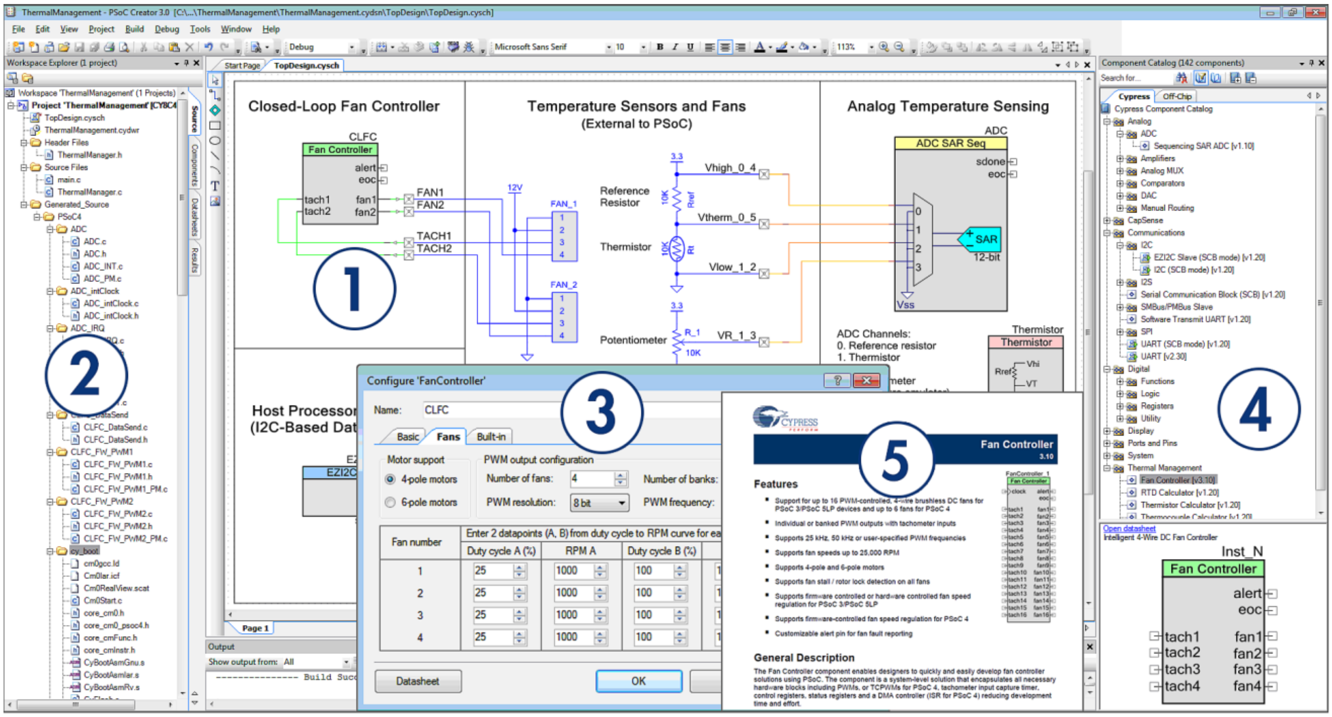
- **Overview:** PSoC Portfolio, PSoC Roadmap
- **Product Selectors:** PSoC 1, PSoC 3, PSoC 4, or PSoC 5LP. In addition, PSoC Creator includes a device selection tool.
- **Datasheets**: Describe and provide electrical specifications for the PSoC 4000, PSoC 4000S, PSoC 4100S, PSoC 4100, and PSoC 4200,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



ThermalManagement - PSoC Creator 3.0
Workspace Explorer (1 project)
Closed-Loop Fan Controller
Temperature Sensors and Fans
(External to PSoC)
Analog Temperature Sensing
Fan Controller
ADC SAR Seq
Host Processor
(I2C-Based Dat
Component Catalog (142 components)
1
2
3
4
5

Configure 'FanController'

Name: CLFC

Basic | Fans | Built-in

Motor support: ● 4-pole motors ○ 6-pole motors

PWM output configuration

Number of fans: 4

PWM resolution: 8 bit

Fan number	Duty cycle A (%)	RPM A	Duty cycle B (%)
1	25	1000	100
2	25	1000	100
3	25	1000	100
4	25	1000	100

Datasheet | OK

Document History

Document Title: PSoC4 CapSense® Low Power One Button – CE210291

Document Number: 002-10291

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**		DCHE	10/08/2015	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 is a registered trademark 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
SanJose, CA 95134-1709

Phone : +1-408-943-2600
Fax : +1-408-943-6848
Website : www.cypress.com