

Objective

This code example demonstrates how to use the SENT TX Component to send CapSense® data over Single Edge Nibble Transmission (SENT) protocol.

Overview

This code example scans a capacitive slider and a button and sends the touch status over the SENT protocol. It also sends the touch status over the I²C bus, so that you can view the touch status using CapSense Tuner in PSoC® Creator™.

Requirements

Tool: PSoC Creator 4.1. See also [Upgrade Information](#).

Programming Language: C (ARM® GCC 5.4-2016-q2-update, ARM MDK 5.22)

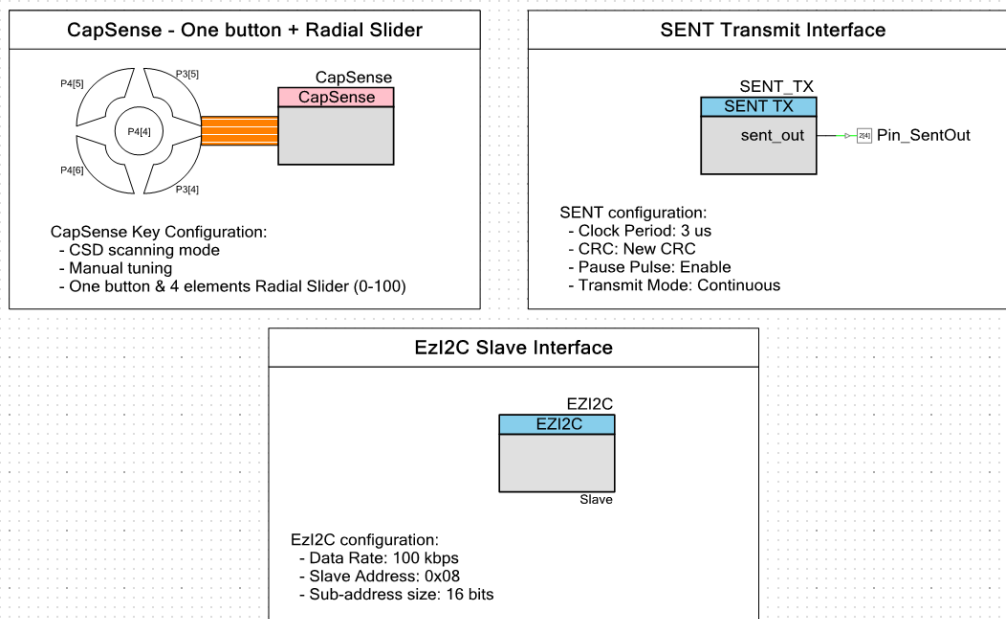
Associated Parts: All PSoC® 4 parts

Related Hardware: [CY8CKIT-042](#), [CY8CKIT-042-BLE-A](#), [CY8CKIT-044](#), [CY8CKIT-046](#), [CY8CKIT-145-40XX](#)

Design

This design uses the CapSense Component to scan the radial slider (made of four capacitive sensors in a circular fashion) and the center capacitive button based on CY8CKIT-044 - PSoC 4 M-Series Pioneer Kit. Raw data of these sensor elements are processed for touch detection. The touch status is sent over the SENT protocol using the SENT_TX Component. The CapSense tuner is also configured to send the touch status over the I²C protocol using the EZI2C Component.

Figure 1. Project Schematic Based on CY8CKIT-044 - PSoC 4 M-Series Pioneer Kit



Design Considerations

This code example works as it is with CY8CKIT-044 PSoC 4 M-Series Pioneer Kit. For other kits, configure the slider to radial or linear in the CapSense Component and select the pins accordingly in the .cydwr file.

Hardware Setup

Connect the SENT receiver input or oscilloscope probe to pin P2[4] to monitor the SENT transmitter output.

Software Setup

The default configuration of PSoC Creator is sufficient for this code example.

Components / User Modules

Table 1 lists the PSoC Creator Components used in this example, and the hardware resources used by each. Pin assignment for capacitive sensors vary based on the kit as shown below:

Table 1. List of PSoC Creator Components

Component	Version	Hardware Resources (For CY8CKIT-044)
CapSense	4.0	One CapSense block, 2 IDACs
SENT_TX	1.0	One TCPWM
EZI2C	3.20	One SCB
Digital Output Pin	2.20	Three I/Os: One for SENT output (P2[4]) Two for EZI2C – SDA (P4[1]), SCL (P4[0]) – Hidden in TopDesign
Analog Pin	2.20	Six I/Os: (all are hidden in TopDesign) Four for Radial slider (P3[5], P3[4], P4[6], P4[5]) One for Button (P4[4]) One for Cmod (P4[2])

Component	Version	Hardware Resources (For CY8CKIT-042)
Analog Pin	2.20	Six I/Os: (all are hidden in TopDesign) Four for Linear slider (P1[1], P1[2], P1[3], P1[4]) One for Button (P1[5]) One for Cmod (P4[2])
Please change Radial Slider to Linear Slider in the CapSense component and change line 149 in main.c file to below: <code>centroid_position = (uint8)CapSense_GetCentroidPos(CapSense_LINEARSLIDER0_WDGT_ID);</code>		

Component	Version	Hardware Resources (For CY8CKIT-042-BLE-A)
Analog Pin	2.20	Six I/Os: (all are hidden in TopDesign) Four for Linear slider (P2[1], P2[2], P2[3], P2[4]) One for Button (P2[5]) One for Cmod (P4[0])
Please change Radial Slider to Linear Slider in the CapSense component and change line 149 in main.c file to below: <code>centroid_position = (uint8)CapSense_GetCentroidPos(CapSense_LINEARSLIDER0_WDGT_ID);</code>		

Component	Version	Hardware Resources (For CY8CKIT-046)
Analog Pin	2.20	Six I/Os: (all are hidden in TopDesign) Four for Radial slider (P4[5], P4[4], P4[7], P4[6]) One for Button (P0[6]) One for Cmod (P4[2])

Component	Version	Hardware Resources (For CY8CKIT-145-40XX)
Analog Pin	2.20	Six I/Os: (all are hidden in TopDesign) Four for Linear slider (P0[0], P0[1], P0[2], P0[3]) One for Button (P0[6]) One for Cmod (P4[1])
Please change Radial Slider to Linear Slider in the CapSense component and change line 149 in main.c file to below: <code>centroid_position = (uint8)CapSense_GetCentroidPos(CapSense_LINEARSLIDER0_WDGT_ID);</code>		

Parameter Settings

Table 2 lists the configuration settings of the CapSense Component:

Table 2. CapSense Component Configuration

Parameter	Value	Comment
Basic:		
CSD Tuning Mode	Manual Tuning	CapSense Tuner can be used to tune the hardware parameters such as sense clock frequency, IDAC values, and software parameters including different thresholds
Widget 1	RadialSlider0 – Self-cap (4 sensing elements)	Configure the Radial Slider to use four sensing elements
Widget 2	Button0 – Self-cap (1 sensing element)	
Advanced (General):		
Enable IIR filter (First order)	Enable	Enable the IIR filter to reduce noise
Advanced (CSD Settings):		
Modulator clock frequency (kHz)	24000	Default
Inactive sensor connection	Ground	Default
IDAC sensing configuration	IDAC sourcing	Default
Enable IDAC auto-calibration	Disabled	Disable IDAC auto-calibration so it can also be tuned using CapSense tuner
Enable compensation IDAC	Enable	Default
Advanced (Widget Details):		
RadialSlider0	Scan resolution – 9 bits Modulator IDAC – 70	Based on manual tuning
Compensation IDAC of different sensor elements of Radial Slider0	RadialSlider0_Sns0 – 40 RadialSlider0_Sns1 – 42 RadialSlider0_Sns2 – 43 RadialSlider0_Sns3 – 50	Based on manual tuning
Button0 (CSD)	Sense clock frequency – 6000 kHz Scan resolution – 10 bits Modulator IDAC – 60	Based on manual tuning
Compensation IDAC of Button0	Button0_Sns0 – 55	Based on manual tuning
Threshold parameters are set to default		

Table 3 lists the configuration settings of the SENT_TX Component:

Table 3. SENT_TX Component Configuration

Parameter	Value	Comment
Clock Period	3 μ s (Default)	Tick time
CRC	New CRC (Default)	New CRC as defined in SAE J2716 (Revised 2016-04)
Pause Pulse	Enable	Pause pulse is enabled in the SENT packet
Transmit Mode	Continuous (Default)	SENT component will keep transmitting last packet

Table 4 lists the configuration settings of the EZI2C Component:

Table 4. EZI2C Component Configuration

Parameter	Value	Comment
Data rate (kbps)	100	100 kbps I2C data rate
Primary slave address (7-bits)	0x08	I2C Slave address = 0x08
Sub-address size (bits)	16	Sub-address = 16 bits

Design-Wide / Global Resources

Figure 2 shows the pin-out assignment in .cydwr file

Figure 2. Pin out assignment

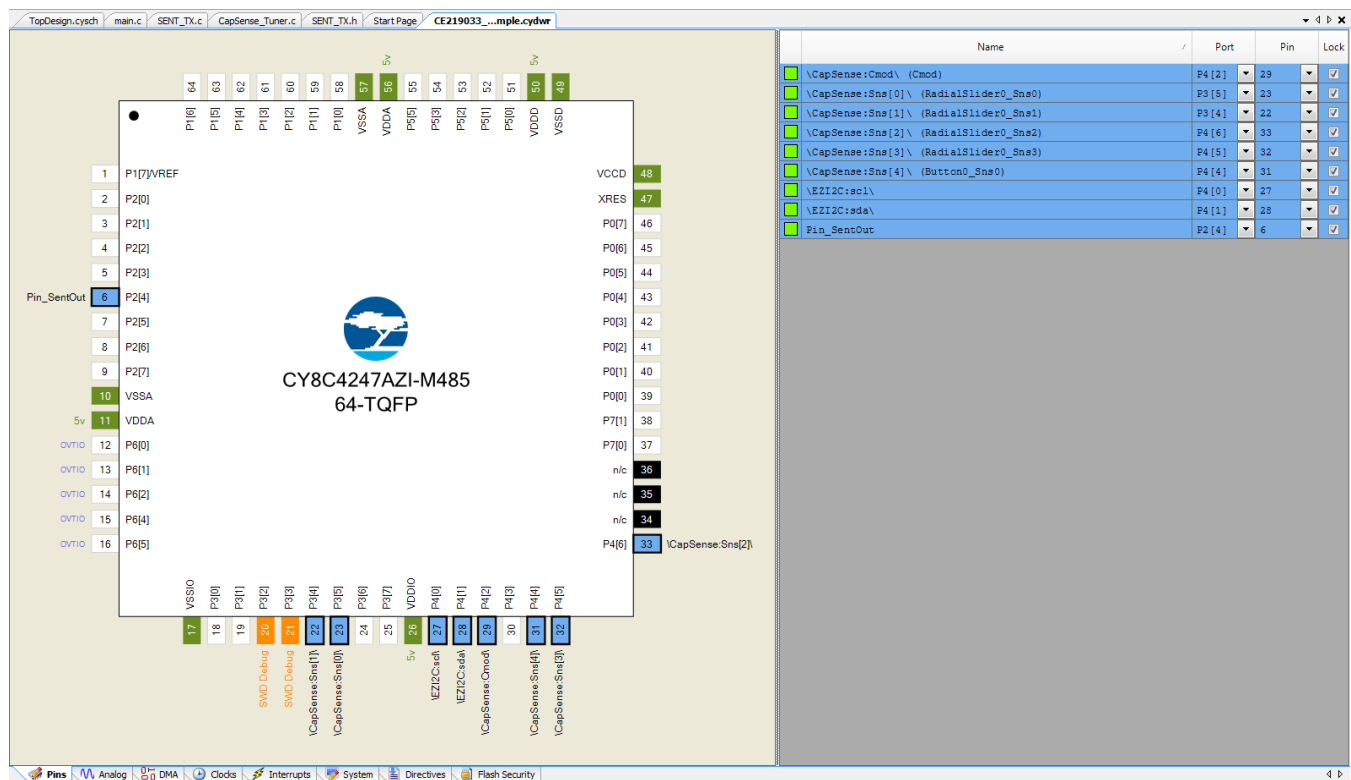


Figure 3 shows how to set the V_{DDA} , V_{DDD} , and V_{DDIO} supply voltages to 5 V. If this is not set to 5 V, the SENT_TX Component will throw an error as per SAE J2716 (Revised 2016-04-29) spec.

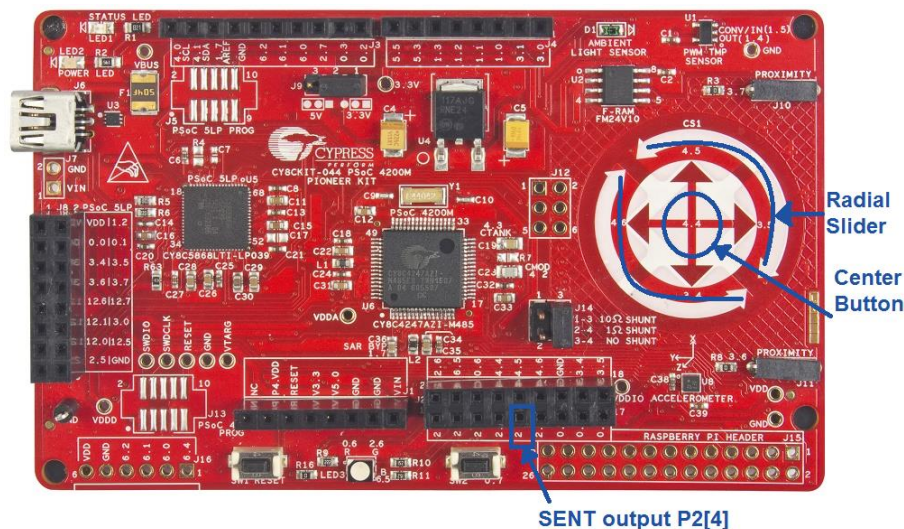
Figure 3. Operating Conditions (Supply Voltages)

Operating Conditions	
VDDA (V)	5
VDDD (V)	5
VDDIO (V)	5
Variable VDDA	<input checked="" type="checkbox"/>

Operation

Figure 4 shows how to test this code example with CY8CKIT-044 – PSoC 4200M Pioneer Kit.

Figure 4. Testing with CY8CKIT-044 - PSoC 4200M Pioneer Kit



To evaluate this code example, perform the following steps:

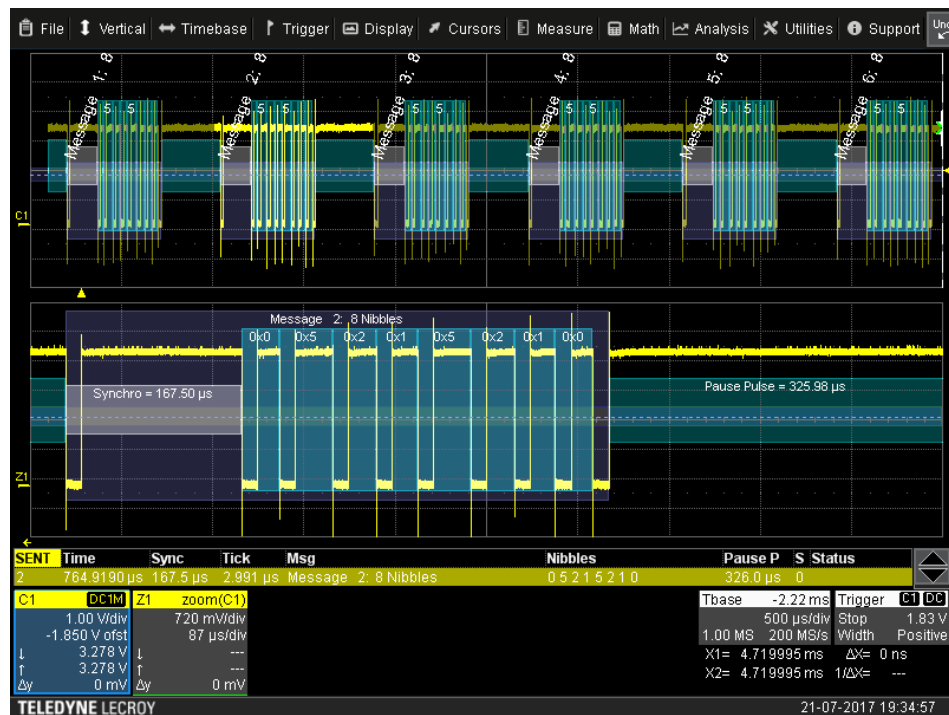
- Connect the SENT decoder or oscilloscope to monitor the SENT output pin P2[4]. Figure 5 shows the SENT output decoded using SENT decoder in LeCroy WaveSurfer 10 oscilloscope.
 - Data Nibble 0 (Status) = 0
 - Data Nibble 1 (Most significant nibble of slider position) = 0xF (No touch on slider = 0xFF)
 - Data Nibble 2 (Least significant nibble of slider position) = 0xF
 - Data Nibble 3 (Button status) = 0x0 (No touch on button = 0x00)
 - Data Nibble 4 = Data Nibble 1
 - Data Nibble 5 = Data Nibble 2
 - Data Nibble 6 = Data Nibble 3
 - Data Nibble 7 (Calculated checksum)
- Place the finger on the radial slider to see the change in data nibbles 1-2 and 4-5.
- Place the finger on the center button to see the change in data nibbles 3 and 6.

Figure 6 shows the SENT output when the finger is touching both radial slider and the center button.

Figure 5. SENT decoder output (LeCroy WaveSurfer 10 oscilloscope) for No-Touch condition



Figure 6. SENT Decoder Output for Touch Detection on Slider and Button



Upgrade Information

Not required.

Related Documents

Application Notes	
AN64846	Getting Started with CapSense®
AN79953	Getting Started with PSoC® 4
AN85951	PSoC® 4 and PSoC Analog Coprocessor CapSense® Design Guide
PSoC Creator Component Datasheets	
TCPWM	PSoC 4 Timer Counter Pulse Width Modulator (TCPWM)
CapSense	PSoC 4 Capacitive Sensing (CapSense®)
EZI2C	PSoC 4 Serial Communication Block (SCB) configured as EZI2C Slave
Pins	Supports connection of hardware resources to physical pins
Device Documentation	
PSoC 4 Datasheets	PSoC 4 Technical Reference Manuals
Development Kit (DVK) Documentation	
PSoC 4 Kits	

Document History

Document Title: CE219033 – SENT_TX Basic Code Example

Document Number: 002-19033

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5874630	KIKU	09/22/2017	New spec

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
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6](#)

Cypress Developer Community

[Forums](#) | [WICED IOT Forums](#) | [Projects](#) | [Videos](#) | [Blogs](#) | [Training](#) | [Components](#)

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.



© Cypress Semiconductor Corporation, 2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spanion 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, Spanion, the Spanion 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.