CY3280-28XXX Universal CapSense® Controller Development Kit Guide

Spec. # 001-57457 Rev. **

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone (USA): 800.858.1810
Phone (Intnl): 408.943.2600
http://www.cypress.com

Copyrights

PSoC® and CapSense® are registered trademarks and CapSense Plus™ and PSoC Designer™ are trademarks of Cypress Semiconductor Corporation (Cypress), along with Cypress® and Cypress Semiconductor™. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Purchase of I^2C components from Cypress or one of its sublicensed Associated Companies conveys a license under the Philips I^2C Patent Rights to use these components in an I^2C system, provided that the system conforms to the I^2C Specification as defined by NXP.

The information in this document is subject to change without notice and should not be construed as a commitment by Cypress. While reasonable precautions have been taken, Cypress assumes no responsibility for any errors that may appear in this document. No part of this document may be copied, or reproduced for commercial use, in any form or by any means without the prior written consent of Cypress. Made in the U.S.A.

Disclaimer

CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Flash Code Protection

Note the following details of the Flash code protection features on Cypress devices.

Cypress products meet the specifications contained in their particular Cypress Data Sheets. Cypress believes that its family of products is one of the most secure families of its kind on the market today, regardless of how they are used. There may be methods that can breach the code protection features. Any of these methods, to our knowledge, would be dishonest and possibly illegal. Neither Cypress nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Cypress is willing to work with the customer who is concerned about the integrity of their code. Code protection is constantly evolving. We at Cypress are committed to continuously improving the code protection features of our products.

Contents

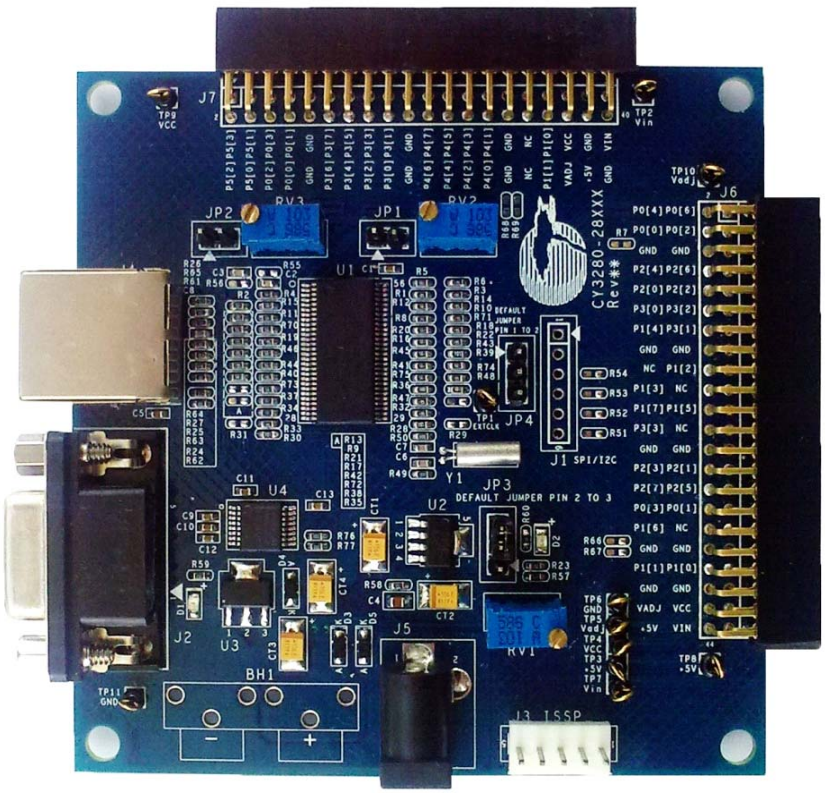
1. Introduction

Welcome to the CY3280-28XXX Universal CapSense® Controller Board. This kit is designed to show the features of CY8C28XXX. The CY8C28XXX family of PSoC® 1 devices includes the following parts: CY8C28403-24PVXI, CY8C28413-24PVXI, CY8C28513-24AXI, CY8C28623-24LTXI, CY8C28433-24PVXI, CY8C28533-24AXI, CY8C28243-24PVXI, CY8C28643-24LTXI, CY8C28445-24PVXI, CY8C28545-24AXI, and CY8C28645-24LTXI. The CY3280-28XXX Universal CapSense Controller Board contains the 56-pin OCD part. The 56-pin OCD part is only used for in-circuit debugging.

Note OCD parts are NOT available for production.

The CapSense feature of CY8C28XXX can be implemented with the CY3280-SLM Universal CapSense Linear Slider Module. The two boards are connected by a 44-pin connector. The other features of CY3280-28XXX can be implemented with the CY3280-CPM1 Universal CapSense Plus™ Controller Module. The two boards are connected by a 40-pin connector. The CY3280-SLM Universal CapSense Linear Slider Module and the CY3280-CPM1 Universal CapSense Plus Controller Module can be connected to the CY3280-28XXX Universal CapSense Controller Board at the same time.

Figure 1-1. CY3280-28XXX Universal CapSense Controller Board



1.1 Kit Components

The following items are included in the kit:

- CY3280-28XXX Universal CapSense Controller Board
- CY3280-28XXX Universal CapSense Controller Board CD
- CY3240-I2CUSB Board
- CY3210-MiniProg1 Programmer
- PSoC Designer 5.0 SP6 CD
- Printed Documents

These tools are not included in CY3280-28XXX Universal CapSense Controller Board Kit, but may be needed to use it. They can be found in other Cypress kits. These kits are available at www.cypress.com. To demonstrate CapSense functions, the kit requires a CY3280-SLM to work. To demonstrate CapSense Plus functions, the kit requires a CY3280-CPM1

- CY3280-SLM Universal CapSense Linear Slider Module Board
- CY3280-CPM1 CapSense Plus Module for CY8C28XXX
- CY3215-ICE
- CY3250-28XXX POD

1.2 CD Directory Structure

This list describes the higher level directory structures in the CD-ROM and does not explore the lower level directories.

|---Docs	'Docs' contains the kit documentation in PDF form
|	
|---Hardware	'Hardware ' contains the design files used in the development of the Kit
| |---Schematic	
| |---BOM	
| |---SilkScreen	
| |---Gerber	
|	
|---Firmware	'Firmware' contains the firmware related files
| |-- CapSense Example Project	
| |-- CapSense Plus Example Projects	
|	
|---Software	Software' contains PC software tools
| |---PSoC Tools	
| |---I2CUSB Command	
|	

1.3 Board Features

- Supports CY3280-SLM Universal CapSense Linear Slider Module
- Supports CY3280-CPM1 CapSense Plus Module
- External 12V power supply or 9V battery input
- Fixed 5V or adjustable voltage for PSoC
- ICE interface for in-circuit debugging
- MiniProg programmer interface
- RS232 interface
- LEDs indicate power status
- 32.768 kHz external OSC option
- Multiple test points

1.4 Documentation Conventions

Table 1-1. Document Conventions for Guides

Convention	Usage
`Courier New`	Displays file locations, user entered text, and source code: `C:\ ...cd\icc\`
Italics	Displays file names and reference documentation: Read about the *sourcefile.hex* file in the *PSoC Designer User Guide*.
[**Bracketed, Bold**]	Displays keyboard commands in procedures: [**Enter**] or [**Ctrl**] [**C**]
File > Open	Represents menu paths: File > Open > New Project
Bold	Displays commands, menu paths, and icon names in procedures: Click the **File** icon and then click **Open**.
Times New Roman	Displays an equation: 2 + 2 = 4
Text in gray boxes	Describes Cautions or unique functionality of the product.

1.5 Document Revision History

Table 1-2. Revision History

Revision	PDF Creation Date	Origin of Change	Description of Change
**	01/18/10	JPLU	Initial Release

2. Quick Start

In this chapter you will learn to get started with the CY3280-28XXX Universal CapSense® Controller Development Kit, use the board as programmed by the factory, and create a CapSense project using PSoC® Designer™.

2.1 Getting Started

This section contains instructions for installing PSoC Designer software and a discussion of CapSense best practices.

2.1.1 Install PSoC Designer Development Software

1. Insert the PSoC Designer Development Software 5.0 Installation CD.
2. Install PSoC Designer Development Software 5.0.
3. Install .NET Framework 2.0.
4. Install PSoC Programmer.
5. A C Compiler license is required to build PSoC Designer C Language projects.

2.1.2 CapSense Best Practices

The Universal CapSense Controller was created using the best practices for CapSense layout. To enable universality and development of the kit and its projects, certain design elements have been changed from what is recommended for final products. Table 2-1 is a list of the design features in the Universal CapSense Controller and what to change for final products.

Table 2-1. Design Features in Universal CapSense Controller

Design Features	Reason for Feature	Impact	Recommended Change
Sensing traces are routed through a connector to sensors.	Buttons, sliders, and LEDs are placed on the module board for greater flexibility with custom modules for development and subsequent releases.	Connectors increase the parasitic capacitance of the sensors, effectively reducing their sensitivity. Connectors also create another path for noise to enter the system.	Sensors and control circuitry should be located on the same printed circuit board. Lower parasitic capacitance by reducing trace lengths.
Sensing traces are routed to other schematic elements.	Universality of the board is enabled by population/ depopulation of 0Ω resistors	Solder pads of 0Ω resistors increase parasitic capacitance.	Route traces directly to sensing elements. Use as few 0Ω resistors as possible.
Sensing traces are located on the top layer.	Using vias to route traces to bottom of board and back to connector increases parasitic capacitance.	Possible noise sensitivity to stimulus on top side of board. Finger presses on routing of control board can lead to sensor activation.	Route sensing traces on non-user side of printed circuit board. Route sensing traces as far from noise sources as possible.
Several regulators are used, including a variable regulator.	To demonstrate CapSense at several voltages.	Global and User Module parameters may need to be verified with changing power supply.	Supply one regulated voltage to PSoC.
Test point on CMOD.	Accessibility of charge/ discharge waveforms	A test point increases noise sensitivity by acting as an antenna.	Solder-pad test points for leads offer better noise immunity if test points are required.
dGND spacing is generalized for noise immunity and sensitivity.	Universality of kit required middle-ground on many parameters.	Design is not optimized for high noise or very thick overlays	Increase spacing for thicker overlays and better sensitivity. Decrease spacing for greater noise immunity.
Connection to shield electrode is through a jumper (module -J2)	Flexibility of module boards for both CSD and CSA control boards.	Higher resistance paths can impair performance of shield electrode in CSD projects.	Dedicated trace for shield electrode. Remove jumpers wherever possible.

Table 2-1. Design Features in Universal CapSense Controller *(continued)*

Design Features	Reason for Feature	Impact	Recommended Change
ESD protection circuitry is not included.	Development/evaluation platform without consistent overlay is inherently vulnerable to ESD events	Direct or air-separated ESD testing may impair operation or damage circuitry. ±2 kV limit on PSoC pins (see data sheet).	Include an overlay and ESD protection circuitry.
User Module Parameters set to supplied overlay thicknesses.	Projects optimized for supplied hardware.	Sensitivity may not be high enough for very thick overlays.	Thicker overlays may require verification of parameters to ensure proper operation.
Unused pins are not routed directly to GND.	Pins brought out to connector for subsequent modules or custom designs.	Possible noise path.	Tie unused sensing traces directly to ground.
0Ω resistors populated throughout.	Universality of the board enabled by population/ depopulation of 0Ω resistors	Solder pads of 0Ω resistors increase parasitic capacitance.	Route traces directly to sensing elements. Use as few 0Ω resistors as possible

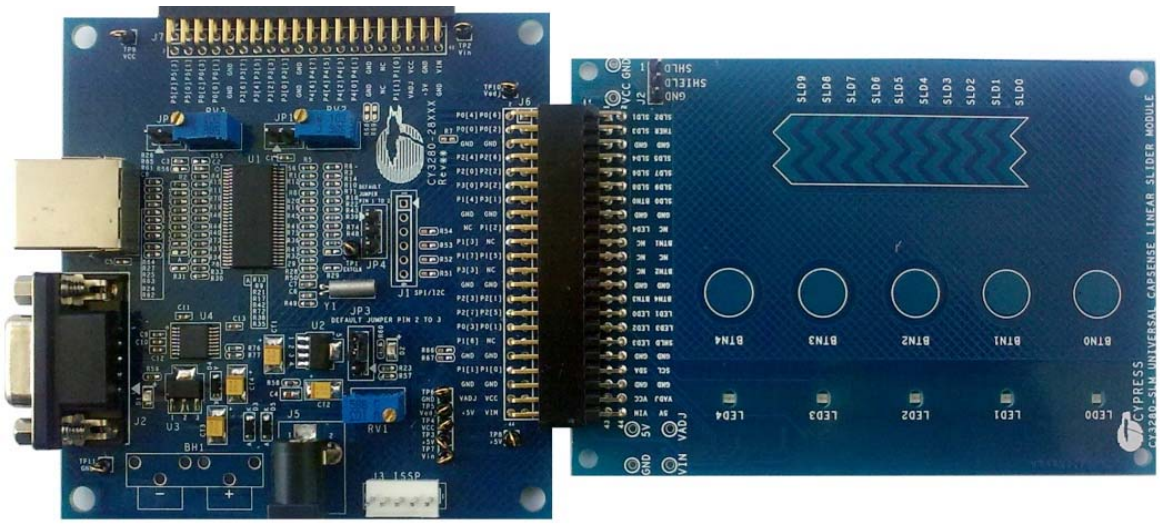
2.2 Using the Board as Factory Programmed

The CY3280-28XXX board is preprogrammed with demonstration firmware. When powered by a PSoC MiniProg, a CY3240-I2USB Bridge, or an optional external power supply, the LEDs light up when a finger touches one of the buttons.

These instructions assume that your board has not been reprogrammed from the factory settings. If it has, and you would like to follow along with this demonstration, follow the instructions in section 2.2.3 Resetting the Board to the Original Factory Programming on page 12. Start this example with 2.2.1 Powering the Board.

2.2.1 Powering the Board

1. Connect CY3280-SLM Universal CapSense Linear Slider Module Board.



2. Place shunts on pin2 and 3 of JP3 (Default Setting).
3. Connect your computer to the CapSense test board ISSP Connector (J3) using the PSoC MiniProg and a USB cable. If this is your first time using the MiniProg, you will need to install the driver using these steps before proceeding:
   a. When the Found New Hardware Wizard opens, select **Install the software automatically (Recommended)** option and click **Next**.
   b. A warning message may tell you that the software you are trying to install has not passed Windows Logo testing. Click **Continue Anyway** each time it appears.
   c. When the installation is complete, click **Finish**.
4. Open PSoC Programmer by going to the Windows Start menu and selecting All Programs > Cypress > PSoC Programming 3.10 > PSoC Programmer.
5. From the Port menu, select **MiniProg1/<Identification Code>**
6. Click **Toggle Device Power**. The D1 and D2 LEDs on the CY3280-28XXX board light red.

2.2.2 Testing the Board

Touch one or more buttons with your finger. The LEDs light up corresponding to the buttons being pressed.

Figure 2-1. CapSense Buttons

2.2.3 Resetting the Board to the Original Factory Programming

Follow these steps to reset the board to the original factory installed programming:

1. Place shunts on pins 2 and 3 of JP1 and pins 1 and 2 of JP4. (Default Setting)
2. To reset the board to the factory conditions, connect your computer to the CY3280-28XXX board ISSP Connector (J3) using the PSoC MiniProg and a USB cable.
3. Open PSoC Programmer by going to the Windows Start menu All Programs > Cypress > PSoC Programming 3.10 > PSoC Programmer.
4. Click **File Load**, navigate to, and open the *CY3280_28XXX_slm.hex* file on the CD at: `..\Firmware\Capsense Example Project\CY3280_28XXX_SLM\`
5. From the Device Family menu, select **CY8C28XXX**.
6. From the Device menu, select **CY8C28645-24LTXI**.
7. Click **Program**. "Programming Succeeded..." appears in the Actions pane when programming is complete.

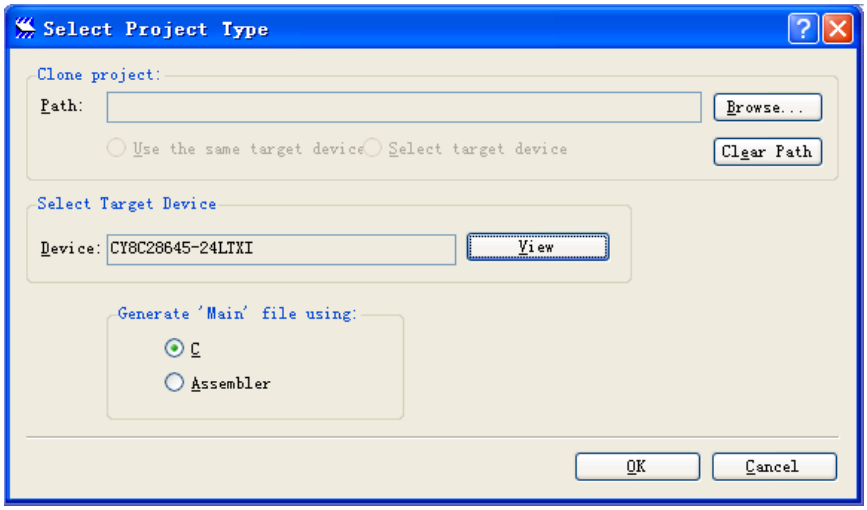
2.3 Creating a CapSense Project Using PSoC Designer 5.0

This section walks you through the steps of creating a PSoC Designer project from scratch. At the end of the project, you will be able to touch a button on the board and see the corresponding LED light up.

2.3.1 Starting a New Project

1. Open PSoC Designer 5.0.
2. Select File > New Project.
3. Name the project **MyProject**.
4. If needed, click **Browse** to save the project in a different location.
5. Click **OK**.
6. Click **View** button, and select **CY8C28645-24LTXI** and click **Select** button.
7. Select **C language** for 'Main' file generation. Click **OK**.

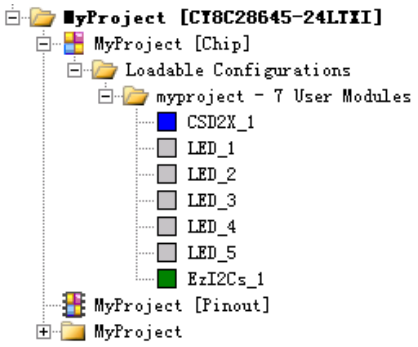
Figure 2-2. Select Project Type Window



2.3.2 Adding CSD2X, LED, and EzI2Cs User Modules to Your Design

1. In the User Module window, expand Cap Sensors folder. Double click **CSD2X** or right click **CSD2X** and select **Place**.
2. Place CSD2X UM with default configuration.
3. Expand Misc Digital folder. Double click **LED** or right click **LED** and select **Place**.
4. Repeat until there are five LEDs in your design.
5. Expand Digital Comm. Double click **EzI2Cs** or right click **EzI2Cs** and select **Place**.
6. All seven selected user modules should show in the Workspace Explorer window.

Figure 2-3. Workspace Explorer

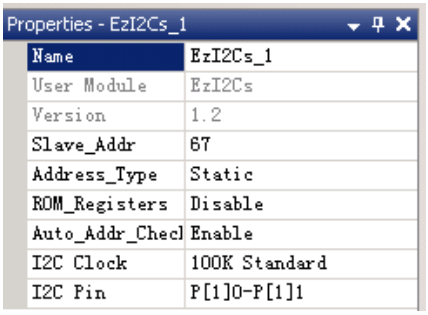


2.3.3 Configuring LED and EzI2Cs User Modules

1. Select **LED_1** user module in the Workspace Explorer window.
2. Set the port to **Port_1**, the pin to **Port_1_2**, and Drive to **Active Low** in the Properties - LED_1 window.
3. Select **LED_2** user module in the Workspace Explorer window.
4. Set the port to **Port_0**, the pin to **Port_0_3**, and Drive to **Active Low** in the Properties - LED_2 window.
5. Select **LED_3** user module in the Workspace Explorer window.
6. Set the port to **Port_0**, the pin to **Port_0_1**, and Drive to **Active Low** in the Properties - LED_3 window.
7. Select **LED_4** user module in the Workspace Explorer window.
8. Set the port to **Port_2**, the pin to **Port_2_7**, and Drive to **Active Low** in the Properties - LED_4 window.
9. Select **LED_5** user module in the Workspace Explorer window.
10. Set the port to **Port_2**, the pin to **Port_2_5**, and Drive to **Active Low** in the Properties - LED_5 window.

User Module	Port	Pin	Drive
LED_1	Port_1	Port_1_2	Active Low
LED_2	Port_0	Port_0_3	Active Low
LED_3	Port_0	Port_0_1	Active Low
LED_4	Port_2	Port_2_7	Active Low
LED_5	Port_2	Port_2_5	Active Low

11. Select **EzI2Cs_1** user module in the Workspace Explorer window.
12. Set the Slave Addr to **67**, and Address_Type to **Statics**.
13. Set ROM_Registers to **Disable**.
14. Set Auto_Addr_Check to **Enable**.
15. Set the I2C Clock to **100K Standard**.
16. Set the I2C Pin to **P1[0]-P1[1]**.



2.3.4 Configuring the CSD2X User Module

1. Right click the **CSD2X_1** user module, and select **CSDADC Wizard**.
2. Set the Switches to **5**.
3. Assign the Switches to Pin Number., as in the following table.

Switch Number	Pin Number
SW0	P2[3]
SW1	P2[1]
SW2	P3[3]
SW3	P1[3[
SW4	P3[1]

4. Click **OK**. The Wizard will close.
5. In the Properties Windows of CSD2X, set the values as shown in Figure 2-4.

Figure 2-4. CSD2X User Module Configuration

Name	CSD2X_1
User Module	CSD2X
Version	1.0
FingerThreshold	120
NoiseThreshold	80
BaselineUpdateThreshol	200
Sensors Autoreset	Disabled
Hysteresis	10
Debounce	3
NegativeNoiseThreshold	20
LowBaselineReset	50
Scanning Speed	Fast
Resolution	10
IDAC_Value_LEFT	250
IDAC_Value_RIGHT	250
IDAC Range	x32
Reference_LEFT	VBG
Reference_RIGHT	VBG
Ref Value	2
PRS_Polynomial	Short
ShieldElectrodeOut	None
Auto Calibration	Disabled

2.3.5 Adding Code to main.c and Building the Project

1. In the Workspace Explorer window, double click **main.c** under **MyProject > Source Files** folder.
2. Select all text in *main.c* and delete it.
3. Add the following code:

```
#include <m8c.h>            /* part specific constants and macros   */
#include "PSoCAPI.h"        /* PSoC API definitions for all User Modules */

WORD I2C_Buf[15];           /* buffer for EzI2C */


void main(void)
{
    BYTE i;

    EzI2Cs_1_SetRamBuffer(sizeof(I2C_Buf),sizeof(I2C_Buf),(BYTE *)I2C_Buf); /*
setup I2C buffer */
    EzI2Cs_1_Start();                               /* I2C start */

    M8C_EnableGInt ;                                /* enable global interrupt */

    CSD2X_1_Start();                                /* CSD2X start */

    CSD2X_1_InitializeBaselines();                  /* initialize the baseline */
    CSD2X_1_SetDefaultFingerThresholds();          /* set default finger threshold
*/

    while(1)
    {
        CSD2X_1_ScanAllSensors();                   /* scan all sensors */
        CSD2X_1_UpdateAllBaselines();               /* update all baseline */

       if( CSD2X_1_bIsSensorActive(0))             /* check the state of button0 */
            LED_1_On();                             /* turn on LED0 */
        else
            LED_1_Off();                            /* turn off LED0 */

       if( CSD2X_1_bIsSensorActive(1))             /* check the state of button1 */
            LED_2_On();                             /* turn on LED1 */
        else
            LED_2_Off();                            /* turn off LED1 */

       if( CSD2X_1_bIsSensorActive(2))             /* check the state of button2 */
            LED_3_On();                             /* turn on LED2 */
        else
            LED_3_Off();                            /* turn off LED2 */

       if( CSD2X_1_bIsSensorActive(3))             /* check the state of button3 */
            LED_4_On();                             /* turn on LED3 */
        else
            LED_4_Off();                            /* turn off LED3 */

       if( CSD2X_1_bIsSensorActive(4))             /* check the state of button4 */
            LED_5_On();                             /* turn on LED4 */
        else
            LED_5_Off();                            /* turn off LED4 */
```

```
        M8C_DisableGInt;                            /* disable global interrupt */
        for(i=0;i<15;i++)
            I2C_Buf[i] = CSD2X_1_waSnsDiff[i];      /* set the buffer */
        M8C_EnableGInt;                             /* enable global interrupt */
    }
}
```

4. Select Build > Generate/Build 'MyProject' Project and verify the compile finishes with no errors.

2.3.6 Programming the CY3280-28XXX Board

1. Connect your computer to the CapSense test board ISSP Connector (J3) using the PSoC MiniProg and a USB cable. If this is your first time using the MiniProg, you will need to install the driver before proceeding. Follow the instructions in 2.2.1 Powering the Board on page 11.
2. In PSoC Designer, select Program > Program Part. The PSoC Programmer application opens
3. From the Port menu, select **MiniProg1/<Identification Code>**.
4. Click **Program**. "Programming Succeeded..." appears in the Actions pane when programming is complete

2.3.7 Testing the Board

1. Click **Toggle Device Power**.
2. Touch one or more buttons with your finger. The associated LEDs light up corresponding to the buttons being pressed.
3. When you are done, click **Toggle Device Power,** and close PSoC Programmer.
4. Return to PSoC Designer and select File > Save Project.

3. Hardware Design Notes

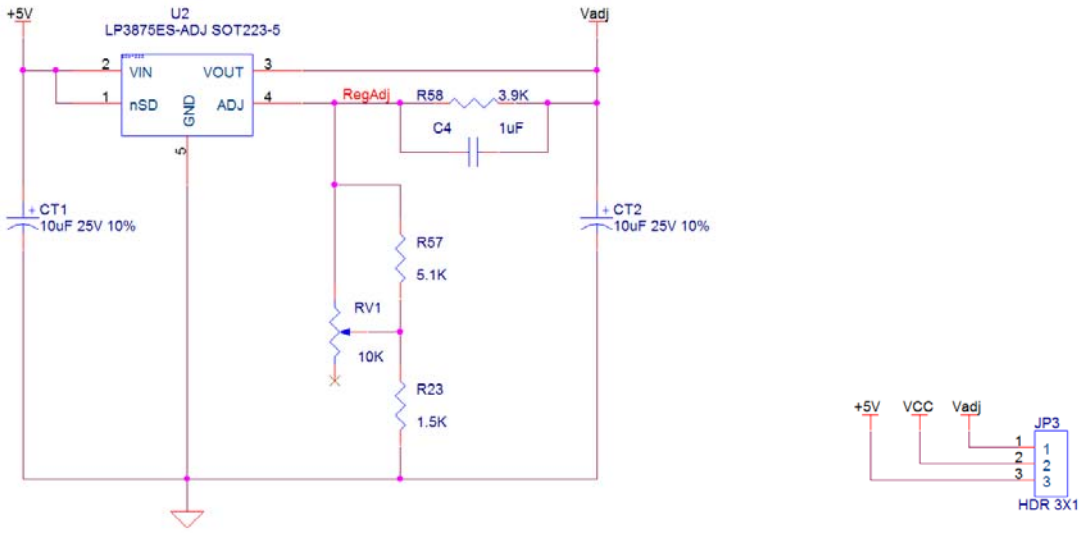
3.1 CY3280-28XXX Universal CapSense® Controller Board Features

The CY3280-28XXX Universal CapSense Controller Board is used to demonstrate the features of the CY8C28XXX family of PSoC® devices. To demonstrate some features of the CY8C28XXX family, a special circuit is needed.

3.1.1 PSoC Power Supplies

There are two kinds of power supplies for PSoC VCC, fixed 5V or variable Vadj. The JP3 is used to select the power supply. If pin 2 is shorted to pin 3, the fixed 5V is selected for the PSoC VCC. If pin 2 is shorted to pin 1, the variable Vadj is selected for the PSoC VCC.

Figure 3-1. Schematics



Vadj is the output of the regulator (LP3875ES-ADJ). According to the schematics, the output can be calculated from the following formula.

$$Vadj = 1.216 \times \left(1 + \frac{R58}{R23 + R57 \parallel RV1}\right)$$

RV1 varies from 0Ω to 10 kΩ, so the maximum voltage and minimum voltage of Vadj can be calculated individually.

$$Vadj_{max} = 1.216 \times \left(1 + \frac{R58}{R23 + R57 \parallel RV1_{min}}\right) = 1.216 \times \left(1 + \frac{3.9k}{1.5k}\right) = 4.38V$$

$$Vadj_{min} = 1.216 \times \left(1 + \frac{R58}{R23 + R57 \parallel RV1_{max}}\right) = 1.216 \times \left(1 + \frac{3.9k}{1.5k + 5.1k \parallel 10k}\right) = 2.19V$$

If you want more information about the LP3875ES-ADJ, refer to the part's data sheet at http://www.national.com/mpf/LP/LP3875-ADJ.html.

If the fixed 5V power supply is selected for the PSoC VCC, follow the steps below.

1. Unplug the external 12V power supply and 9V battery power.
2. Place shunts on pin 2 and pin 3 of JP3.
3. Plug in the external 12V power supply or 9V battery power.

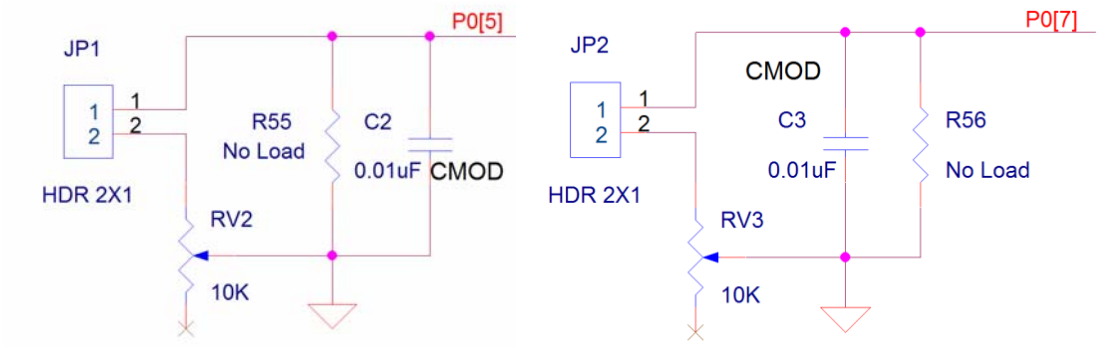
If Vadj is selected for the PSoC power supply, follow the steps below.

1. Remove any shunts on JP3.
2. Plug in the external 12V power supply or 9V battery power.
3. Measure the voltage of pin 1 of JP3 with a multimeter, tuning the potentiometer RV1 until the desired voltage is acquired.
4. Unplug the external 12V power supply and 9V battery power.
5. Place shunts on pin 1 and pin 2 of JP3.
6. Plug in the external 12V power supply or 9V battery power.

3.1.2 Dual Channel CSD Scanning

The most significant improvement of CY8C28XXX over previous parts is the Dual-Channel CSD scanning feature. Demonstrating this new feature requires two external capacitors. According to the *CY8C28xxx PSoC Programmable System-on-Chip Technical Reference Manual*, pin 5 and pin 7 of port 0 are dedicated for the external capacitors' connections. The capacitor Cmod assigned to pin 5 of port 0 is connected to the internal left analog bus. The capacitor Cmod assigned to pin7 of port 0 is connected to the internal right analog bus. You can also use a potentiometer in parallel to Cmod, but this is optional. If the potentiometer RV2 is needed for the left channel, put shunts on J6. If the potentiometer RV3 is needed for the right channel, put shunts on J7. By default, resistors of R55 and R56 are not populated.

Figure 3-2. Schematics

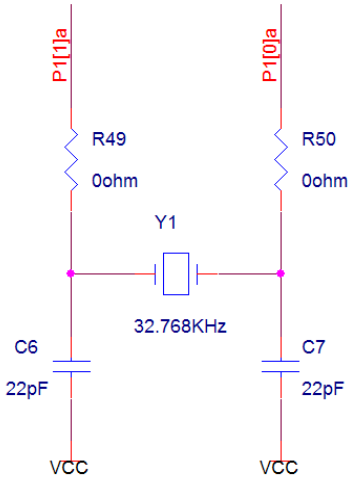


3.1.3 32.768 kHz External Crystal Oscillator

The CY8C28XXX has multiple clock sources. These include the phase locked loop (PLL), internal main oscillator (IMO), internal low speed oscillator (ILO), and 32.768 kHz external crystal oscillator (ECO) for precision, programmable clocking. The clocks, together with programmable clock dividers (as a System Resource), provide the flexibility to integrate almost any timing requirement into the PSoC device.

The external crystal oscillator is assigned to the P1[1] Crystal (XTALin) and P1[0] Crystal (XTALout). By default, the related components are not populated. If the external crystal oscillator is selected as the PSoC clock source, the components R49, R50, C6, C7, and Y1 should be assembled.

Figure 3-3. Schematics



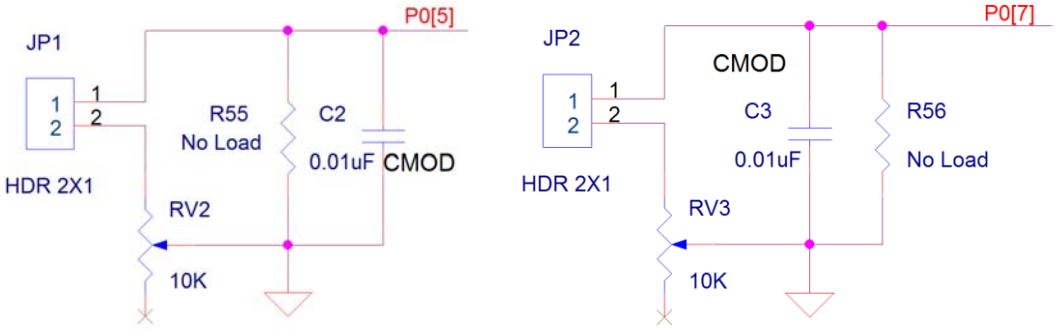
3.2 Hardware Interface and Description

Table 3-1. Connector and Hardware Descriptions

Board ID	Description
JP1	Short to connect potentiometer in parallel with Cmod on left analog bus
JP2	Short to connect potentiometer in parallel with Cmod on right analog bus
JP3	PSoC VCC selection jumper
JP4	XRES selection
BH1	External 9V battery connector
J1	SPI/I2C interface
J2	RS232 interface
J3	ISSP/I2CUSB connector
J4	ICE interface
J5	External power supply
J6	CY3280-SLM board connector
J7	CY3280-CPM1 board connector

3.2.1 Potentiometer in Parallel with Cmod Selection Jumper (JP1, JP2)

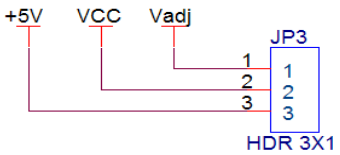
Figure 3-4. Potentiometer in Parallel with Cmod Selection Jumper



If potentiometer RV2 is needed for the left channel, put shunts on JP1. If potentiometer RV3 is needed for the right channel, put shunts on JP2.

3.2.2 PSoC VCC Selection Jumper (JP3)

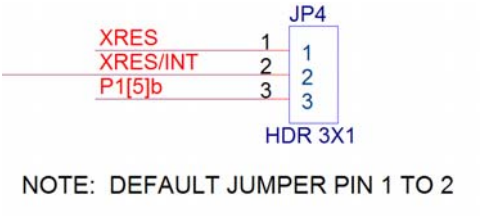
Figure 3-5. PSoC VCC Selection Jumper



Place shunts on pin 2 and pin 3 of JP3 to select fixed 5V as the PSoC VCC power supply. Place shunts on pin 1 and pin 1 of JP3 to select variable Vadj as the PSoC VCC power supply. If VCC power is applied, the power on LED D2 lights up. Refer to PSoC Power Supplies on page 19 for more information.

3.2.3 XRES/INT Selection Jumper (JP4)

Figure 3-6. XRES/INT Selection Jumper



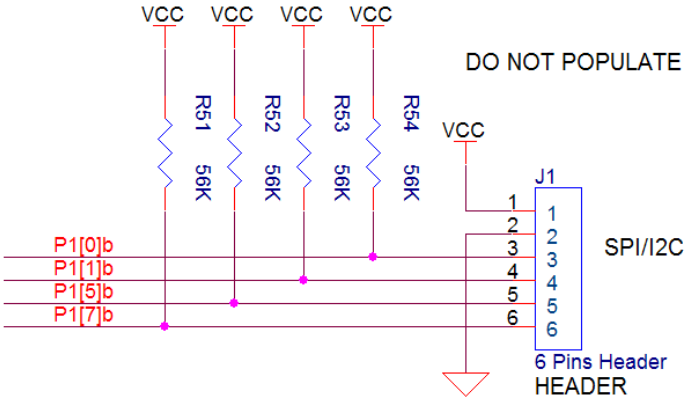
Pin 3 of J3 can serve as the XRES signal or INT signal. If it is the XRES signal, put shunts on pin 1 and pin 2 of JP4. If it is the INT signal, put shunts on pin 2 and pin 3 of JP4. The default setting of JP4 is shorting pin 1 and pin 2.

3.2.4 External 9V Battery Power Connector (BH1)

The board supports a 9V battery power supply. The 9V battery plugs on this connector directly to power the board.

3.2.5 SPI/I2C Interface (J1)

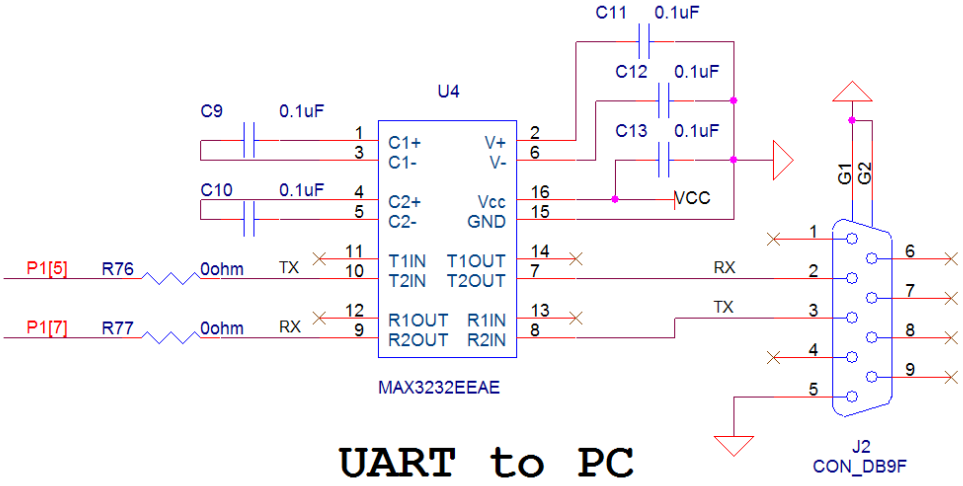
Figure 3-7. SPI/I2C Interface



This interface can be used for debugging, among many other purposes. By default, the related components are not populated. If you want to use this interface, assemble the resistors R29, R31, R35, and R38. R51 to R54 are the pull up resistors for the signal. Assemble any of them as the system requires.

3.2.6 RS232 Interface (J2)

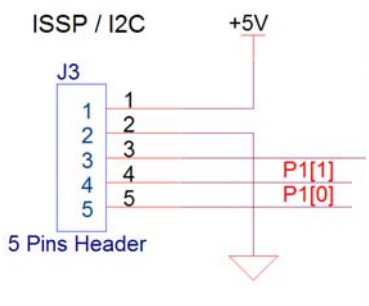
Figure 3-8. RS232 Interface



The J2 connector is an RS232 interface. If this interface is selected, you must assemble resistors R76 and R77. Configure P1[5] to the TX of UART and P1[7] to the RX of UART in the interconnection window of PSoC Designer 5.0.

3.2.7 ISSPI/I2CUSB Interface (J3)

Figure 3-9. ISSPI/I2CUSB Interface



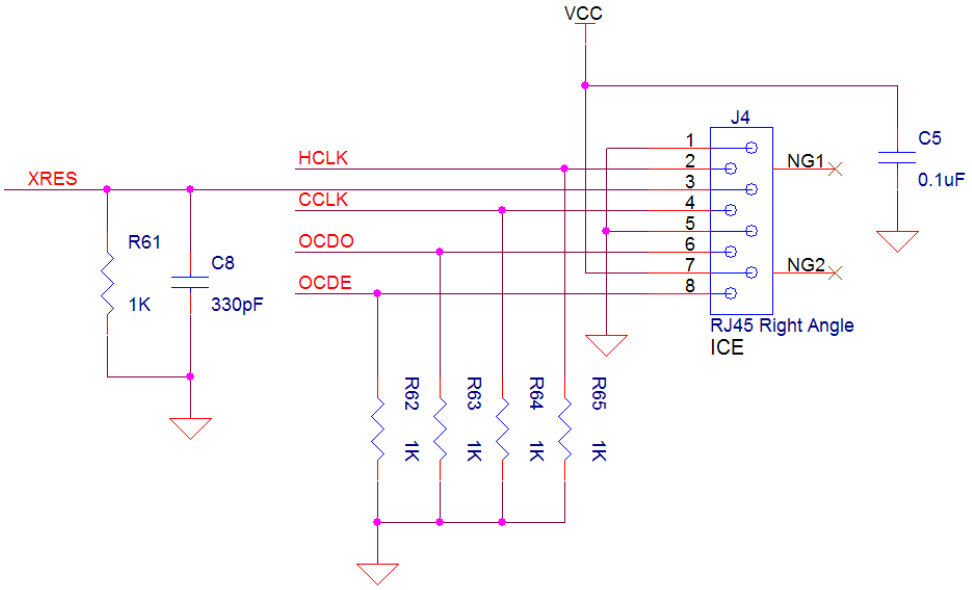
Connector J3 is the ISSP/I2CUSB interface. A PSoC MiniProg programmer can be plugged onto this connector for new code programming.

Table 3-2. Signal Assignment

Pin Number	Signals
1	5V VCC
2	GND
3	XRES/INT
4	SCLK
5	SDATA

3.2.8 ICE Interface (J4)

Figure 3-10. ICE Interface



The CY3280-28XXX Universal CapSense Controller Board has an interface for in-circuit debugging. Plug the CY3250-28XXX POD Emulator Pod in to the CY3215 ICE Cube and connect the pod with J4. The PSoC Designer Debugger provides in-circuit emulation support that allows you to test the

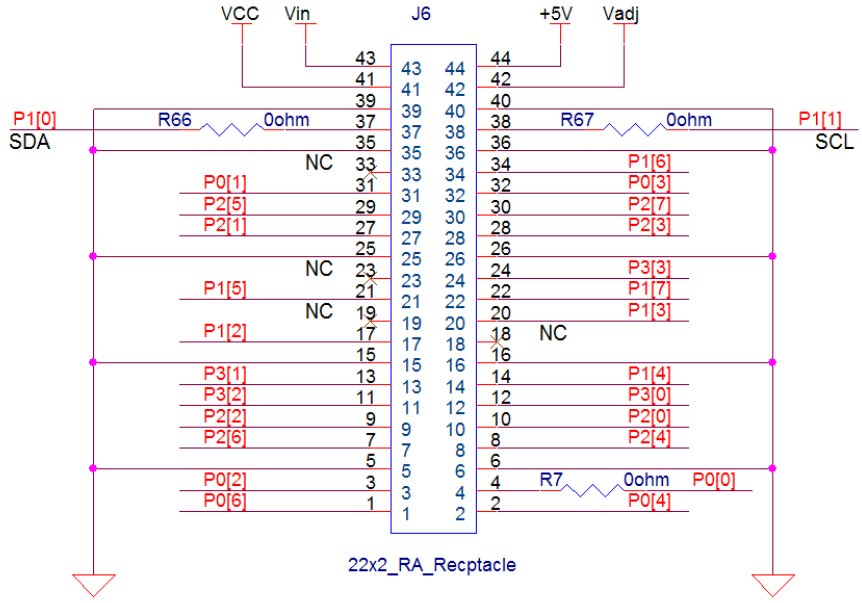
project in a hardware environment while viewing and debugging device activity in a software environment.

3.2.9 External Power Supply Interface (J5)

The J5 connector is the power supply interface. A DC power supply between 9V and 12V is acceptable. The power on LED D1 lights up when power is applied.

3.2.10 CY3280-SLM Board Connector (J6)

Figure 3-11. CY3280-SLM Board Connector



The CY3280-28XXX Universal CapSense Controller Board connects with the CY3280-SLM Universal CapSense Linear Slider Module through the P2 connector. The CY3280-SLM Universal CapSense Linear Slider Module is used for CapSense demonstration. By default, R66, R67, and R7 are not populated. Table 3-3 indicates the usage of pins when these two boards are connected.

Table 3-3. Pin Usage

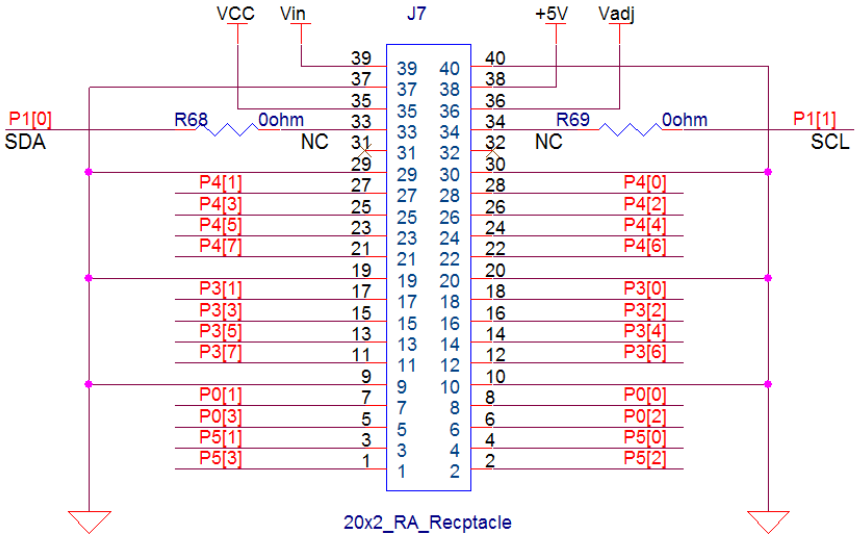
Pin	PSoC Port	Description
1	P0[6]	Connect to Slider9 of CY3280-SLM Board
2	P0[4]	Connect to Slider8 of CY3280-SLM Board
3	P0[2]	Connect to Slider7 of CY3280-SLM Board
4	P0[0]	Connect to AnalogOut of CY3280-SLM Board
5	GND	Ground.
6	GND	Ground.
7	P2[6]	Connect to Slider6 of CY3280-SLM Board
8	P2[4]	Connect to Slider5 of CY3280-SLM Board
9	P2[2]	Connect to Slider4 of CY3280-SLM Board
10	P2[0]	Connect to Slider3 of CY3280-SLM Board
11	P3[2]	Connect to Slider2 of CY3280-SLM Board

Table 3-3. Pin Usage *(continued)*

Pin	PSoC Port	Description
12	P3[0]	Connect to Slider1 of CY3280-SLM Board
13	P3[1]	Connect to Button5 of CY3280-SLM Board
14	P1[4]	Connect to Slider10 of CY3280-SLM Board
15	GND	Ground.
16	GND	Ground.
17	P1[2]	Connect to LED1 of CY3280-SLM Board
18	NC	N/A
19	NC	N/A
20	P1[3]	Connect to Button4 of CY3280-SLM Board
21	P1[5]	N/A
22	P1[7]	N/A
23	NC	N/A
24	P3[3]	Connect to Button3 of CY3280-SLM Board
25	GND	Ground
26	GND	Ground
27	P2[1]	Connect to Button2 of CY3280-SLM Board
28	P2[3]	Connect to Button1 of CY3280-SLM Board
29	P2[5]	Connect to LED5 of CY3280-SLM Board
30	P2[7]	Connect to LED4 of CY3280-SLM Board
31	P0[1]	Connect to LED3 of CY3280-SLM Board
32	P0[3]	Connect to LED2 of CY3280-SLM Board
33	NC	N/A
34	P1[6]	Connect to ShieldOut of CY3280-SLM Board
35	GND	Ground
36	GND	Ground
37	P1[1]	Connect to SDL of CY3280-SLM Board
38	P1[0]	Connect to SDA of CY3280-SLM Board
39	GND	Ground
40	GND	Ground
41	VCC	PSoC VCC power supply
42	Vadj	Variable power supply
43	Vin	External power supply
44	5V	Fixed 5V power supply

3.2.11 CY3280-CPM1 Board Connector (J7)

Figure 3-12. CY3280-CPM1 Board Connector



The CY3280-28XXX Universal CapSense Controller Board connects with the CY3280-CPM1 CapSense Plus Module through P3. The CY3280-CPM1 CapSense Plus Module is used to demonstrate extended features of the CY8C28XXX beyond CapSense. By default, R68 and R69 are not populated. Table 3-4 indicates the usage of pins when the CY3280-CPM1 Universal CapSense Plus Controller Board is connected.

Table 3-4. Pin Usage

Pin	PSoC Port	Description
1	P5[3]	N/A
2	P5[2]	N/A
3	P5[1]	N/A
4	P5[0]	N/A
5	P0[3]	N/A
6	P0[2]	N/A
7	P0[1]	Analog input/output signal of CY3280-CPM1 Board
8	P0[0]	Analog output signal of CY3280-CPM1 Board
9	GND	Ground
10	GND	Ground
11	P3[7]	Button output of CY3280-CPM1 Board
12	P3[6]	SPI clock signal input of CY3280-CPM1 Board
13	P3[5]	Button output of CY3280-CPM1 Board
14	P3[4]	SPI data signal input of CY3280-CPM1 Board
15	P3[3]	N/A
16	P3[2]	N/A
17	P3[1]	N/A
18	P3[0]	N/A

Table 3-4. Pin Usage *(continued)*

Pin	PSoC Port	Description
19	GND	Ground
20	GND	Ground
21	P4[7]	N/A
22	P4[6]	N/A
23	P4[5]	LED6 of CY3280-CPM1 Board
24	P4[4]	LED5 of CY3280-CPM1 Board
25	P4[3]	LED4 of CY3280-CPM1 Board
26	P4[2]	SPI clock signal output and LED3 of CY3280-CPM1 Board
27	P4[1]	LED2 of CY3280-CPM1 Board.
28	P4[0]	SPI data signal output and LED1 of CY3280-CPM1 Board
29	GND	Ground
30	GND	Ground
31	NC	N/A
32	NC	N/A
33	P1[1]	Connect to SDL of CY3280-CPM1 Board
34	P1[0]	Connect to SDA of CY3280- CPM1 Board
35	GND	Ground
36	GND	Ground
37	VCC	PSoC VCC power supply
38	Vadj	Variable power supply
39	Vin	External power supply
40	5V	Fixed 5V power supply

3.3 Test Points

Table 3-5. Test Points and Descriptions

Board ID	Description
TP1	External Clock Input
TP2, TP7	Power Supply of Vin
TP3, TP8	Power Supply of 5V
TP4, TP9	Power Supply of VCC
TP5, TP10	Power Supply of Vadj
TP6, TP11	Ground