CY3235-ProxDet

# CapSense® Proximity Detection Demonstration Kit Guide

Doc. #: 001-67986 Rev. *B

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

© Cypress Semiconductor Corporation, 2011. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, 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 products in life-support systems application implies that the manufacturer assumes all risk of such use, and in doing so indemnifies Cypress against all charges.

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress), and is protected by, and subject to worldwide patent protection (United States and foreign), United States copyright laws, and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and/or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

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.

Use may be limited by and subject to the applicable Cypress software license agreement.

PSoC® and CapSense® are registered trademarks and PSoC Designer™ is a trademark of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

## Flash Code Protection

Cypress products meet the specifications contained in their particular Cypress PSoC Data Sheets. Cypress believes that its family of PSoC 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, unknown to Cypress, 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** **5**

1.1 Kit Contents ....5
1.1.1 Prerequisites ....5
1.1.2 CY3240-I2USB Bridge ....5
1.2 PSoC Designer ....6
1.3 PSoC Programmer ....6
1.4 Bridge Control Panel ....6
1.5 Additional Learning Resources ....6
1.6 Document History ....7
1.7 Documentation Conventions ....7

**2. Getting Started** **9**

2.1 Installation ....9
2.2 PSoC Designer ....13
2.3 PSoC Programmer ....14
2.4 Bridge Control Panel ....14
2.5 Install Hardware ....15

**3. Kit Operation** **17**

3.1 Introduction ....17
3.2 Hardware Requirements ....17
3.3 Programming Device ....17
3.4 Bridge Control Panel ....18

**4. Hardware** **23**

4.1 System Block Diagram ....23
4.2 Functional Description ....23
4.2.1 PSoC CY8C21434-24LFXI ....24
4.2.2 ISSP/I2C Connector ....26

**5. Code Examples** **27**

5.1 Example_CY3235_CSD_PROX_Detector Project ....27
5.1.1 Project Description ....27
5.1.2 Device Configuration ....28
5.1.3 Firmware Architecture ....29
5.1.4 Verify Output ....29

**A. Appendix** **31**

A.1 Schematic ....31
A.2 Board Layout ....32
A.3 Bill of Materials (BOM) ....34

# 1. Introduction

Thank you for your interest in the CY3235 CapSense® Proximity Detection Demonstration Kit. The CY3235-ProxDet kit showcases the proximity sensing capability of CY8C21x34 CapSense Controller family of devices. The kit is designed for easy prototyping of a proximity sensor using CapSense technology. A supporting hardware called I2C-to-USB Bridge is also included in the kit for data acquisition and tuning of the CapSense Controller. This bridge connects to the CY3235-ProxDet board via the I2C interface and to the PC via a USB interface. The kit is configured with the Example_CY3235_CSD_Prox_Detector code example when shipped.

## 1.1 Kit Contents

The CY3235-ProxDet Kit contains:

- Proximity detection demo board with sensing wire
- CY3240-I2USB Bridge board
- USB A to Mini B cable (3 feet)
- Supporting software CD
- CY3235-ProxDet quick start guide
- Free sample silicon (CY8C24894)

Inspect the contents of kit. If any parts are missing, contact your nearest Cypress sales office for assistance.

### 1.1.1 Prerequisites

**CY3217-MiniProg1 Programmer**

The CY3217-MiniProg1 programmer is required to program the PSoC® device and is available for purchase separately at http://www.cypress.com/CY3217-MiniProg1. The MiniProg provides the ability to program any Cypress PSoC device quickly and easily. It is small, compact, and connects to your PC using the provided USB 2.0 cable. During prototyping, the MiniProg can be used as an in-system serial programmer (ISSP) to program PSoC devices on your PCB.

When the MiniProg is connected, you can use the PSoC Programmer software to program (this free software can either be launched from within PSoC Designer or run as a standalone program). PSoC Programmer software can be downloaded from http://www.cypress.com/?rID=38050.

### 1.1.2 CY3240-I2USB Bridge

The I2USB Bridge allows to test and tune the CapSense module in PSoC by bridging the PC's USB port to I2C. Populated with the CY8C24894 PSoC device, I2USB Bridge can be connected through ISSP pins (connector J1) on the proximity detection board.

## 1.2 PSoC Designer

PSoC Designer is the PSoC integrated development environment (IDE) that you can use to customize your PSoC application firmware. The latest PSoC Designer has many new features, bug fixes, and support for new PSoC devices.

## 1.3 PSoC Programmer

The PSoC Programmer software tool offers a simple and user friendly GUI that connects to the programming hardware for downloading hex files into the flash memory of a PSoC device.

## 1.4 Bridge Control Panel

The Bridge Control Panel is used with the CY3240-I2USB Bridge to enable communication with I2C slave devices (here with the CY3235-ProxDet kit). This software is used to configure I2C slave devices as well as acquire and chart the data received from I2C slave devices. The Bridge Control Panel helps in optimizing, debugging, and tuning the target devices.

## 1.5 Additional Learning Resources

Visit http://www.cypress.com for additional learning resources in the form of data sheets, technical reference manual, and application notes.

- CY8C21x34.pdf - This is the PSoC CY8C21x34 data sheet, which contains pin descriptions and other important specifications of this device - http://www.cypress.com/?rID=3345
- CY8C24x94.pdf - This is the PSoC CY8C24x94 data sheet, which contains pin discriptions and other important specifications of this device - http://www.cypress.com/?rID=3371
- CY3235_ProxDet_Layout.zip - http://www.cypress.com/go/cy3235-proxdet
- CY335_ProxDet_Schematic.pdf - http://www.cypress.com/go/cy3235-proxdet
- MiniProg Guide e-Book.pdf (MiniProg users guide and code examples) - This document describes the features and usage of MiniProg device - http://www.cypress.com/?rID=3412
- PSoC Designer Training - This is a web-based course that provides an overview of PSoC and its design tools. It is the first of a multi-part series of modules designed to provide you with the necessary training to enable you to quickly take advantage of PSoC Technology, develop your embedded applications and get to market fast - http://www.cypress.com/psoctraining

## 1.6 Document History

| Revision | PDF Creation Date | Origin of Change | Description of Change |
|---|---|---|---|
| ** | 03/14/11 | ARVM | Initial version of kit guide |
| *A | 06/06/11 | ARVM | Updated schematic |
| *B | 10/14/11 | SRVS | Updated title. Minor content updates |

## 1.7 Documentation Conventions

Table 1-1. Document Conventions for Guides

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

# 2. Getting Started

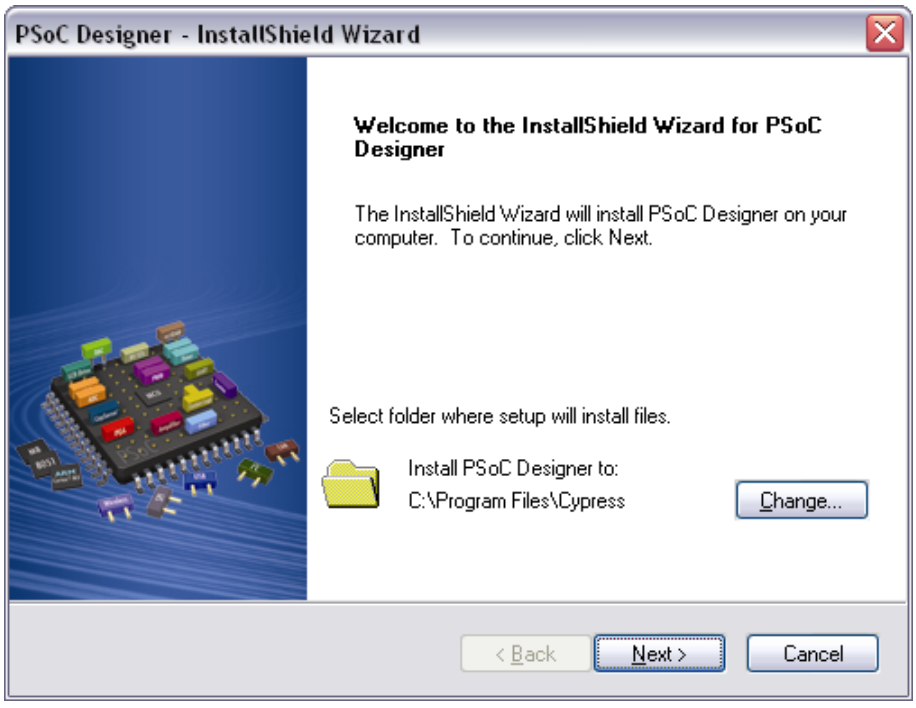
This chapter describes how to install and configure PSoC Designer 5.1.

## 2.1 Installation

To install PSoC Designer 5.1, follow these steps:

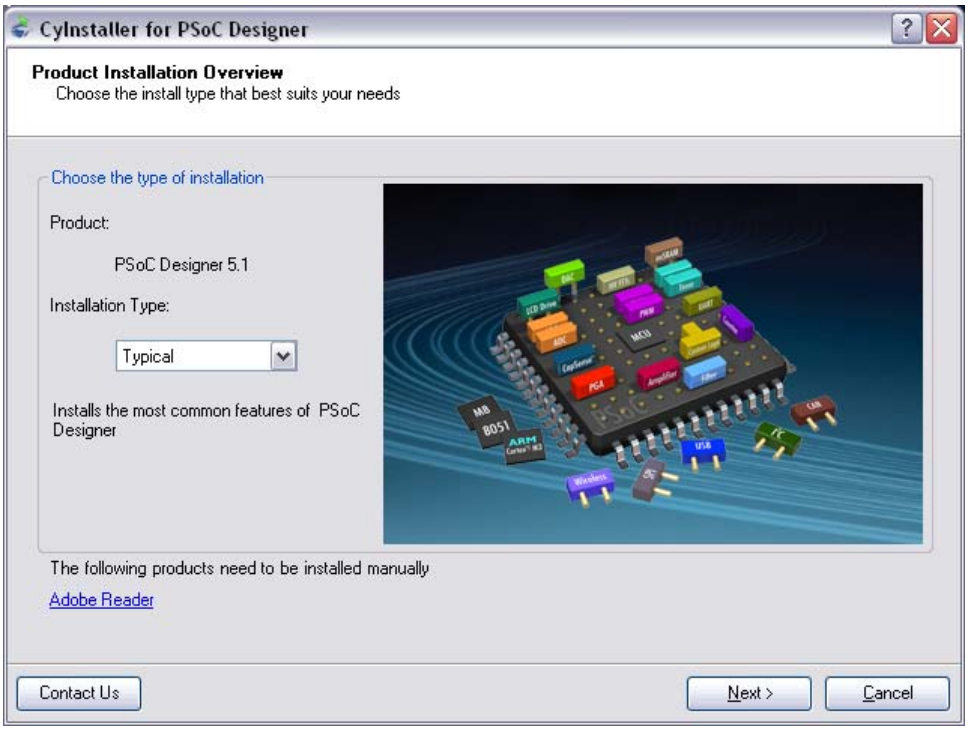
1. Download the PSoC Designer installer file from http://www.cypress.com/?id=2522.
2. Extract the setup file and execute it to start the installation.
3. The InstallShield Wizard screen appears. The default location for setup is shown on the InstallShield Wizard screen. You can change the location for setup using **Change**, as shown in Figure 2-1.
4. Click **Next** to launch the PSoC Designer installer.

Figure 2-1. InstallShield Wizard



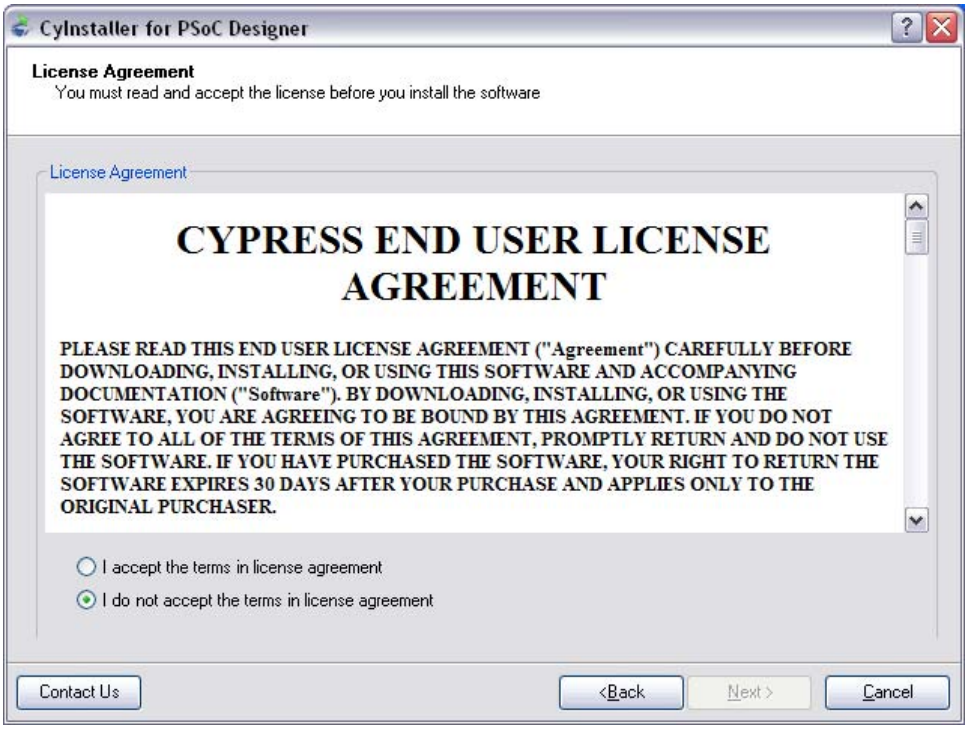

5. On the Product Installation Overview screen, select the installation type that best suits your requirement. The drop-down menu has three options - **Typical, Complete,** and **Custom,** as shown in Figure 2-2.
6. Click **Next**.

Figure 2-2. Installation Type Options



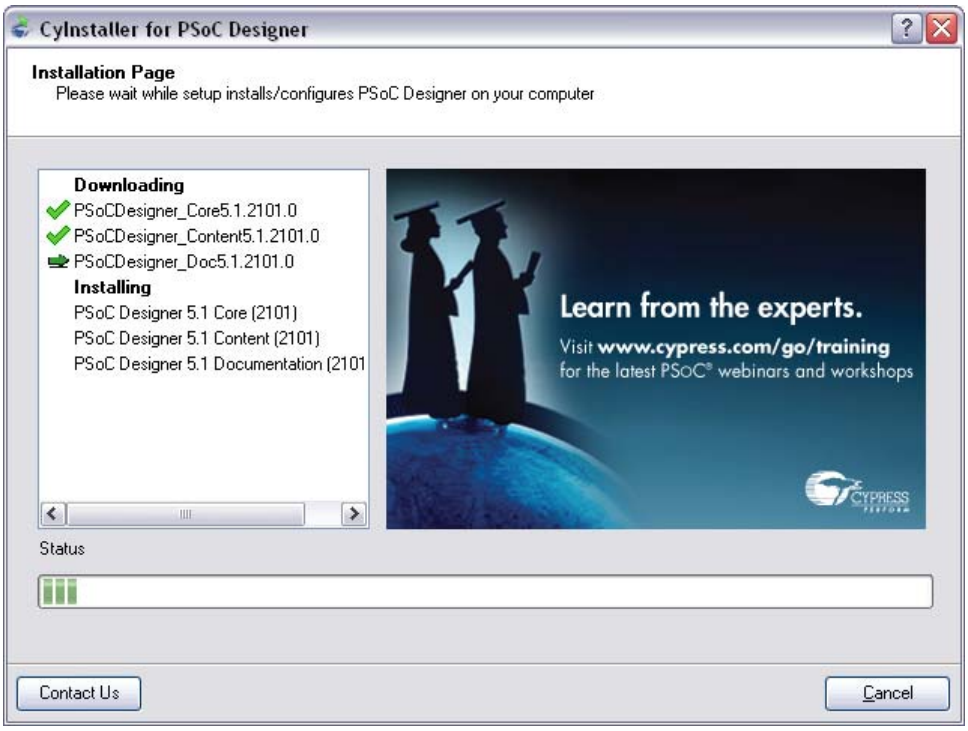

7. The License Agreement screen appears. Read the agreement and click on **I accept the terms in license agreement**.
8. Click **Next**.

Figure 2-3. License Agreement



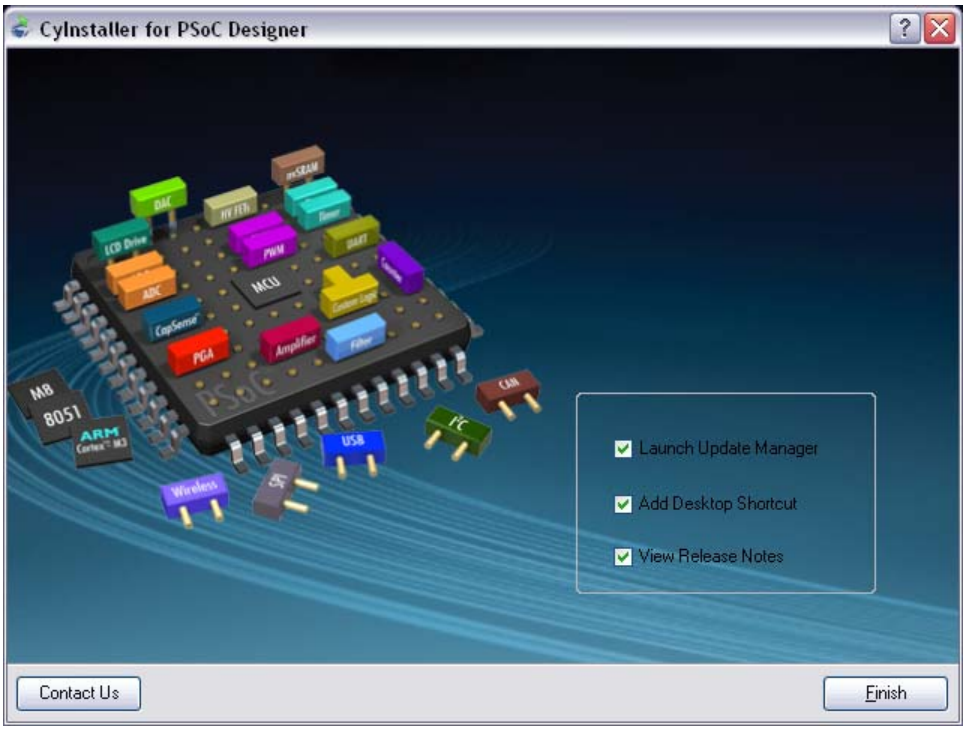
9. When the installation begins, a list of all packages appears on the Installation Page. A green checkmark appears next to every package that is downloaded and installed, as shown in Figure 2-4.
10. Wait until all the packages are downloaded and installed successfully.

Figure 2-4. Installation Page




11. Click **Finish** to complete the PSoC Designer installation.

Figure 2-5. Installation Complete

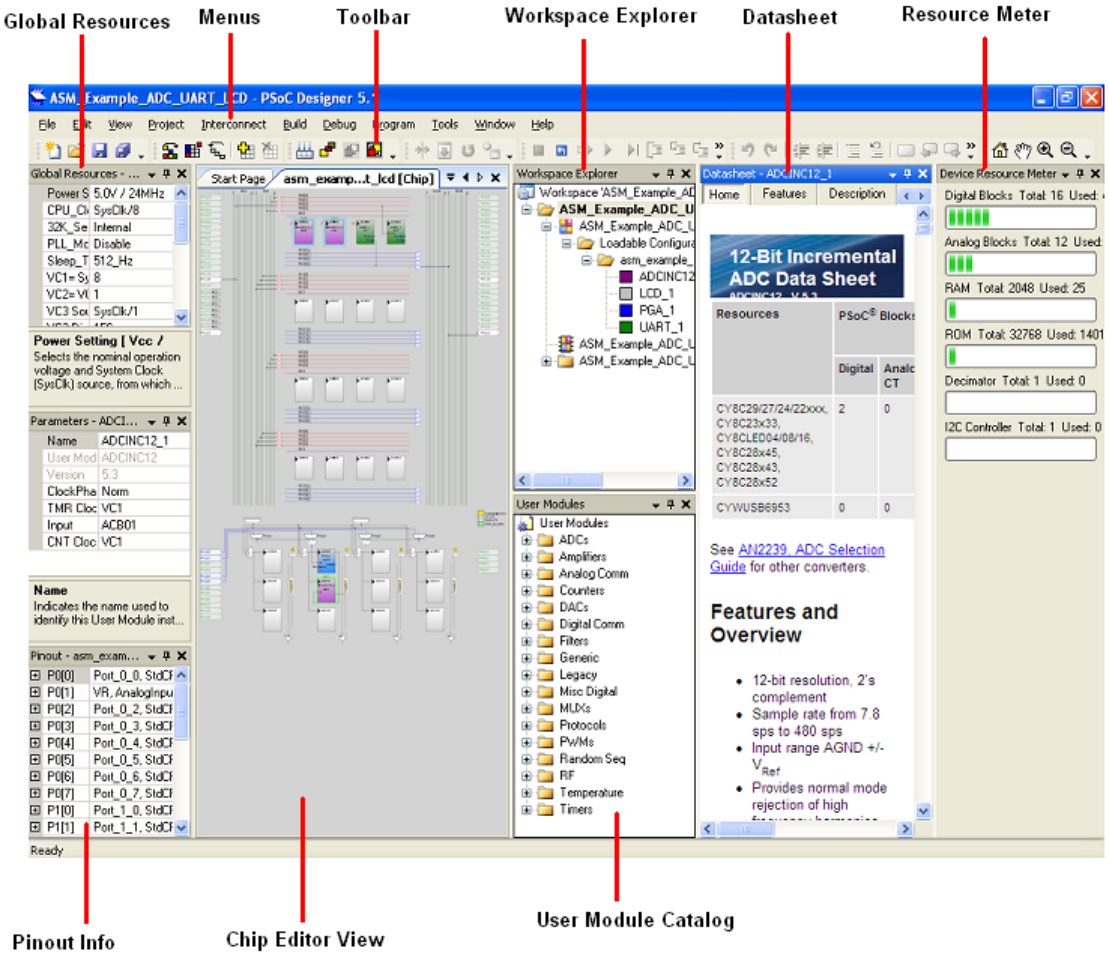



After software installation, verify your installation and setup by opening PSoC Designer, PSoC Programmer, and Bridge Control Panel with the kit board attached over MiniProg and I2USB Bridge.

## 2.2 PSoC Designer

1. Click **Start > Programs > Cypress > PSoC Designer 5.1 > PSoC Designer 5.1**.
2. Click **File > Create New Project**, to create new project; click **File > Open** to work with an existing project.

   Figure 2-6. PSoC Designer Interconnect View



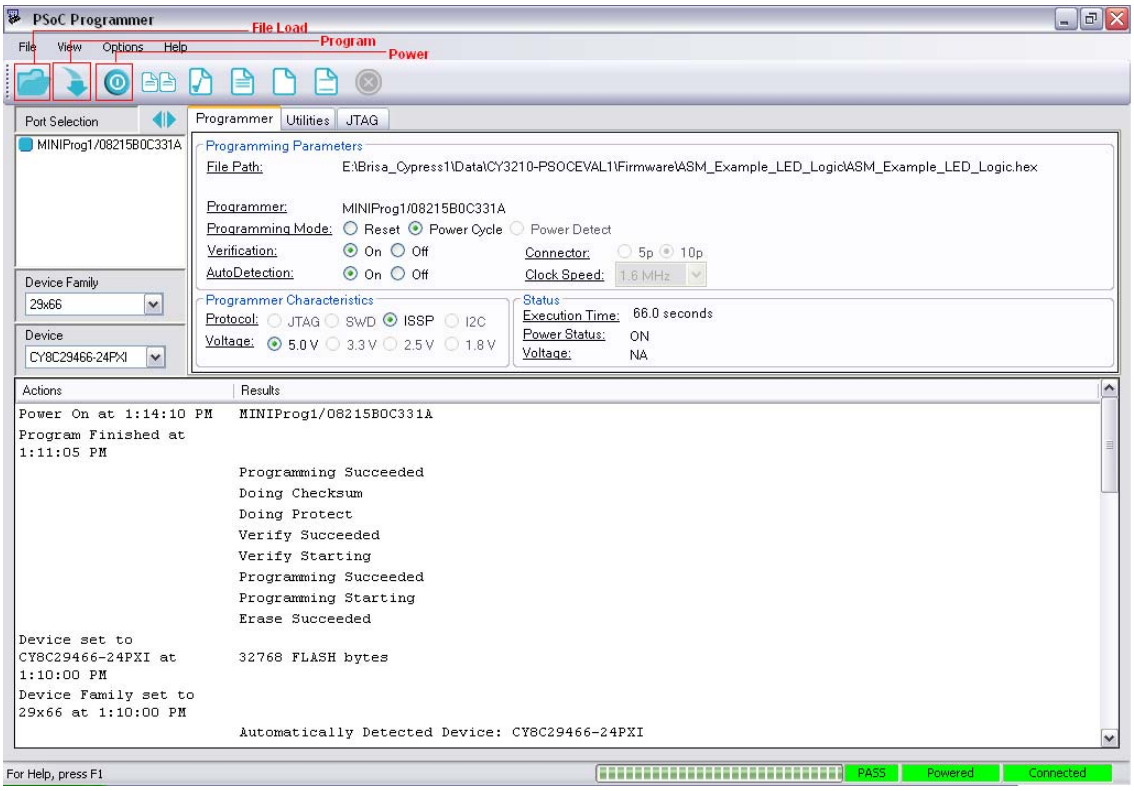

3. To experiment with the code examples, go to Code Examples on page 27.

**Note** For more details on PSoC Designer, go to PSoC Designer IDE Guide at the following location: `<InstallDirectory>:\Program Files\Cypress\PSoC Designer\<version>\Documentation.`

## 2.3 PSoC Programmer

1. Click **Start > Programs > Cypress > PSoC Programmer 3.12 > PSoC Programmer 3.12**.
2. Select MiniProg from the port selection, as shown in Figure 2-7.
3. Use the **Power** button to power up the kit before programming the code on to the chip.

Figure 2-7. PSoC Programmer Window




4. Click **File Load** button to load the hex file.
5. Use **Program** button to program the hex file on to the chip.
6. When programming is successful, **Programming Succeeded** appears in the Actions pane.
7. Close PSoC Programmer.

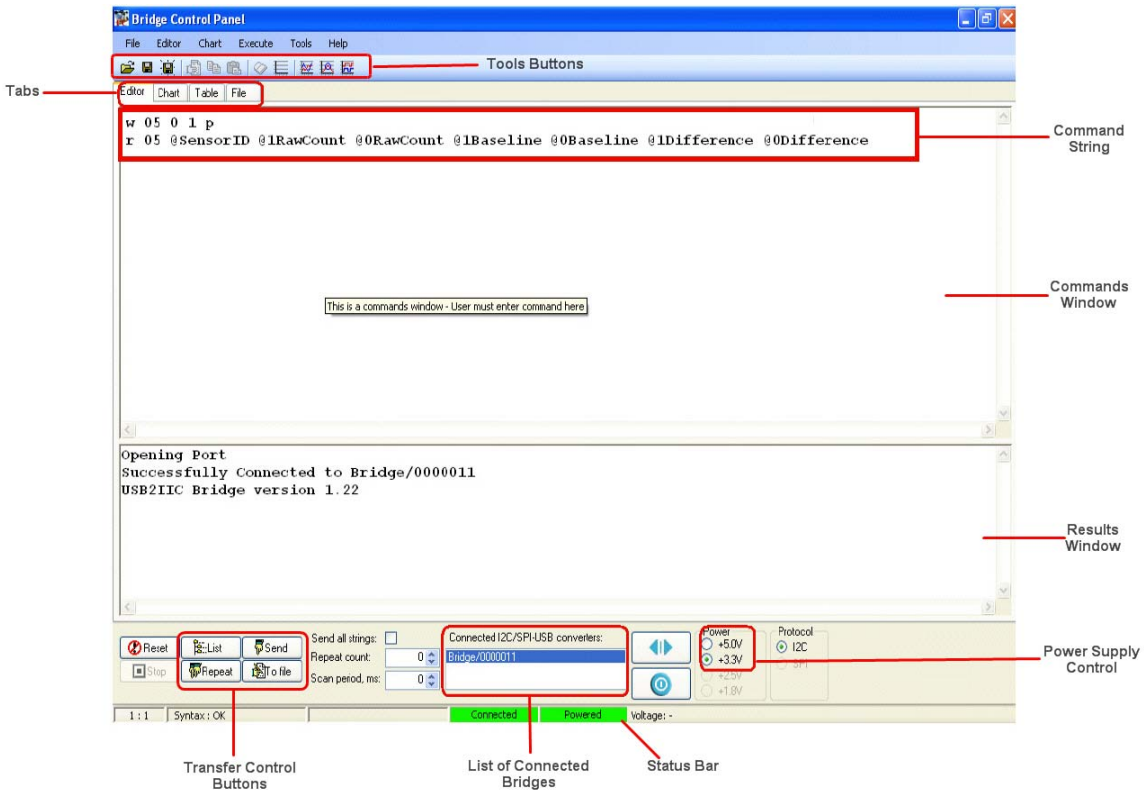
**Note** For more details on PSoC Programmer, see the user guide at the following location: `<InstallDirectory>:\Program Files\Cypress\Programmer\<version>\Documents`.

## 2.4 Bridge Control Panel

1. Click **Start > All Programs > Cypress > Bridge Control Panel 1.2 > Bridge Control Panel 1.2**.
2. Select an I2C-USB bridge and connect to it in the **Connected I2C-USB Converters** window.
3. Power the selected slave device to 5 V or 3.3 V using **Power Supply Control** and **Toggle Power** buttons.
4. Click the **List** button to view the I2C address of the slave device connected to the I2C-USB Bridge. The address is listed in the Results window.
5. Write the I2C commands to be sent to the slave device in the Commands window.
6. Click **Send** to send an I2C command once; click **Repeat** to send the command repeatedly.

7. Define the variable name and type in the Variable Settings window. Then, assign the data bytes read from the slave device to variables in the Commands window.
8. Click the **Variable Settings** option from the Chart menu, click **Load**, navigate to and open a *.ini file to load a saved variable settings file.
9. Click **Open File** from the File menu, navigate to and open a *.iic file to load a saved commands file. The iic file contents appear in the command window of the Bridge Control Panel as highlighted in Figure 2-8.
10. Repeat an I2C read command in which the read data is assigned to variables; click on the **Chart** tab to see a graph of the read data.

Figure 2-8. Selecting Bridge




**Note** For more details on the Bridge Control Panel, go to the I2USB Bridge Help Topics from the Bridge Control Panel menu bar.

## 2.5 Install Hardware

No hardware installation is required for this kit.

# 3. Kit Operation

## 3.1 Introduction

The CY3235-ProxDet board can be connected to I2USB bridge board via the ISSP connector. The kit requires PSoC Designer for firmware development, PSoC Programmer for programming, and Bridge Control Panel to test and tune the CapSense parameters.

## 3.2 Hardware Requirements

- CY3235-ProxDet board (PSoC Device ID: CY8C21434-24LFXI)
- MiniProg programmer
- CY3240-I2USB Bridge
- USB A to Mini-B cable

## 3.3 Programming Device

The CY3235-ProxDet board is programmed using a CY3217-MiniProg.

Figure 3-1. MiniProg Connected to Controller

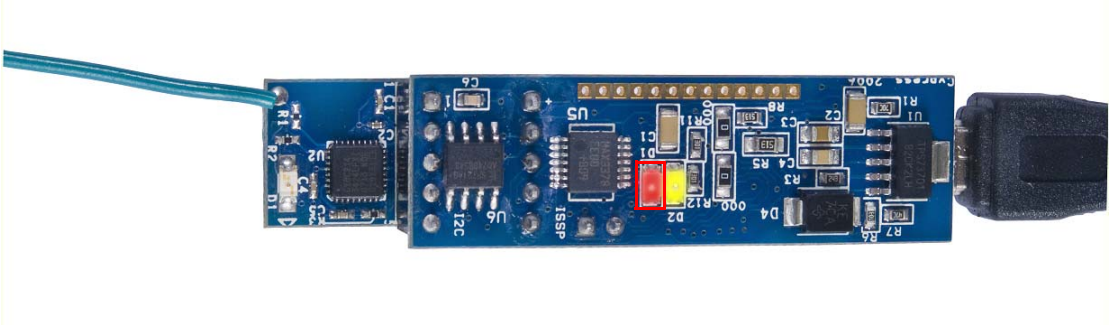


1. Connect the MiniProg to your computer using a USB cable (A to Mini B). Then, connect the Mini-Prog to the proximity detector board's ISSP connector (J1).
2. Program the code example hex file on to the CY3235-ProxDet board using the MiniProg. The hex file is available in the Example_CY3235_CSD_Prox_Detector.zip, which can be downloaded from http://www.cypress.com/go/cy3235-proxdet.
3. While programming is in progress, the target power LED on the MiniProg is on, as marked in Figure 3-1.
4. When **Programming Succeeded** appears in the Actions pane, detach the MiniProg.

## 3.4 Bridge Control Panel

I2USB Bridge is used to get the CapSense data from the controller board. CapSense data of a sensor includes RawCount, Difference count, Baseline, and On/Off status of sensor. This data can be viewed as a live chart using the Bridge Control Panel software. Follow these steps to use Bridge Control Panel with the CY3235-ProxDet board.

Figure 3-2. I2USB Bridge Connected to Controller

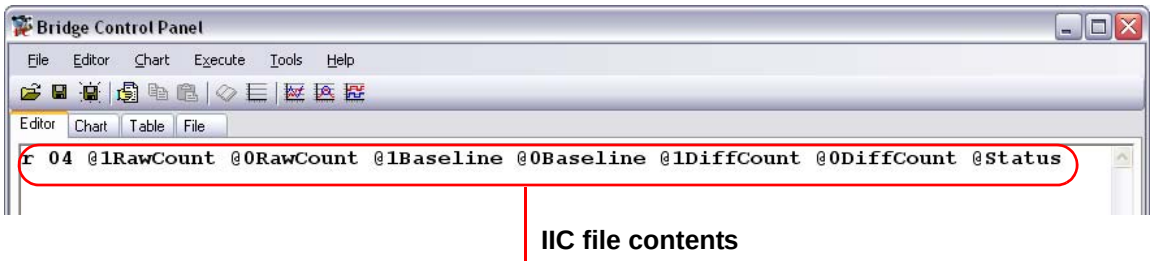


1. Open the Bridge Control Panel.

   **Note** PSoC Designer and PSoC Programmer must be closed before opening the Bridge Control Panel.
2. Connect an I2USB Bridge to the ISSP connector J1 of CY3235-ProxDet board, as shown in Figure 3-2.
3. Connect the I2USB Bridge to your computer through an USB cable.
4. In Bridge Control Panel, the I2USB Bridge ID appears in the Port Selection window. Click on the ID to connect to it.
5. Select **Variable Settings** option from the Chart menu; click **Load**, navigate to and open the Prox-Det.ini file and click **OK**.

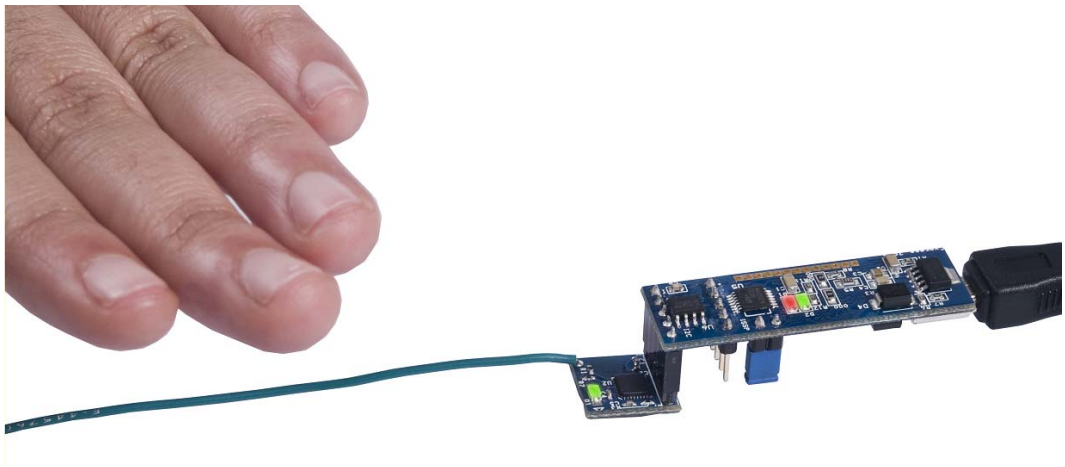
   **Note** The I2C configuration files, *I2USB_Bridge_ConfigFiles.zip* can be downloaded from http://www.cypress.com/?rID=3422
6. Click **Open File** from File menu, navigate to and open the ProxDet.iic file.
7. Select **+5V** in the Power Settings box.
8. Click on **Toggle Power** to power the I2USB Bridge, LED (red) D1 glows, as shown in Figure 3-2.
9. Click on the **List** button to identify the I2C slave address (7-bit address = 0x04).
10. Send the commands in the Editor pane to the controller by pressing the enter key or by clicking on the **Send** button.

Figure 3-3. Bridge Control Panel Editor View



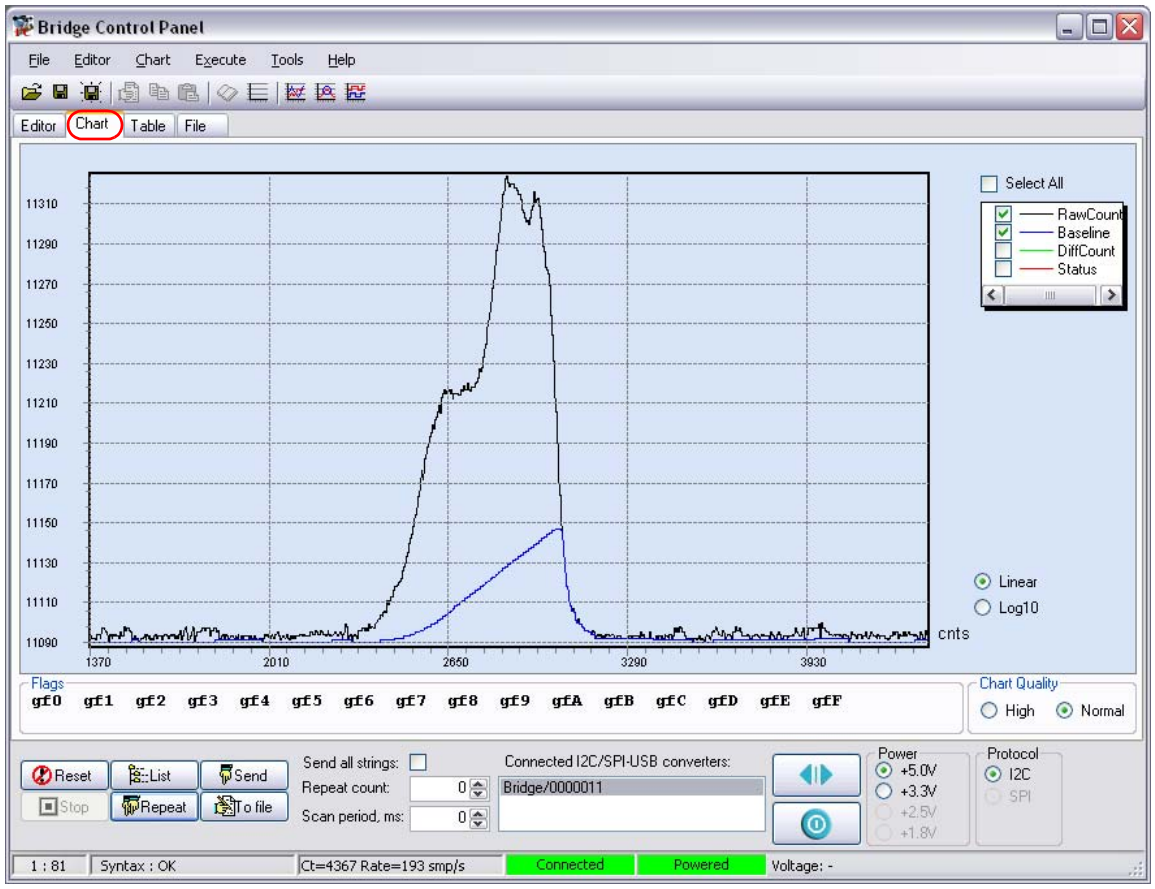

11. Click on **Repeat** to get CapSense data continuously from the controller.
12. Move your hand near the sensing wire of proximity detector board and observe that the LED (D1) glows when the hand is within the detection range of the sensor, see Figure 3-4.

Figure 3-4. LED Glowing on Activation of Proximity Sensor



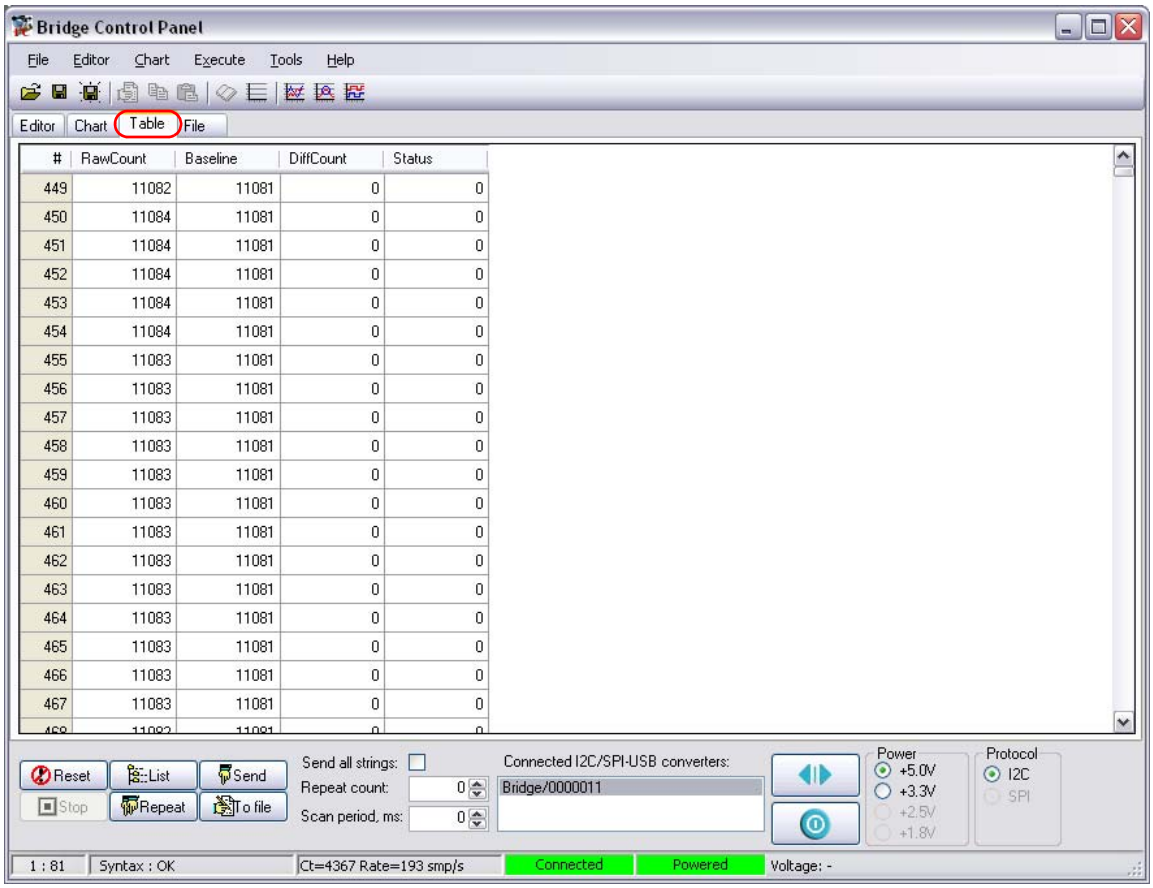
13. Switch to Chart view by clicking on the **Chart** tab, as shown in Figure 3-5. View the respective CapSense data waveforms.

Figure 3-5. Bridge Control Panel Chart View




14. To the right of the Chart view, select the required variables to view, by clicking the corresponding checkbox.
15. Click the **Stop** button to stop scanning.
16. Click the **Table** tab to view the demonstration board variable values, as shown in Figure 3-6.

Figure 3-6. Bridge Control Panel Table View



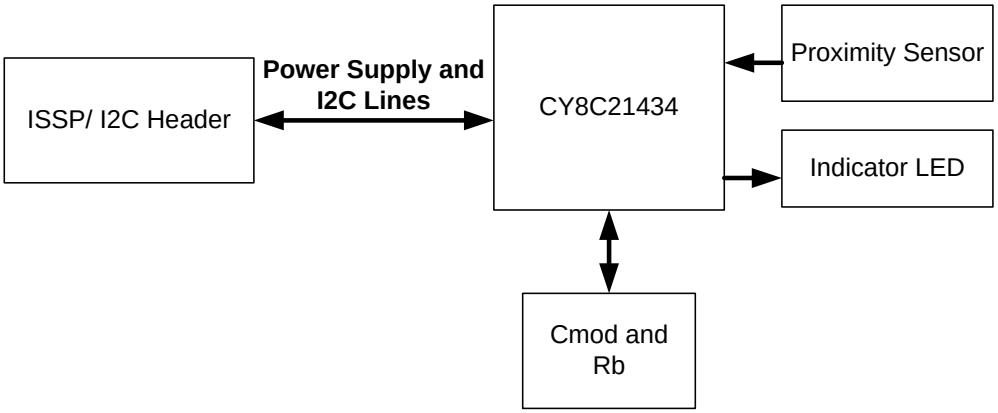
# 4. Hardware

## 4.1 System Block Diagram

The CY3235-ProxDet board has the following sections:

- PSoC CY8C21434-24 LFXI
- ISSP/I2C header
- CMOD and Rb
- Proximity sensor
- Indicator LED

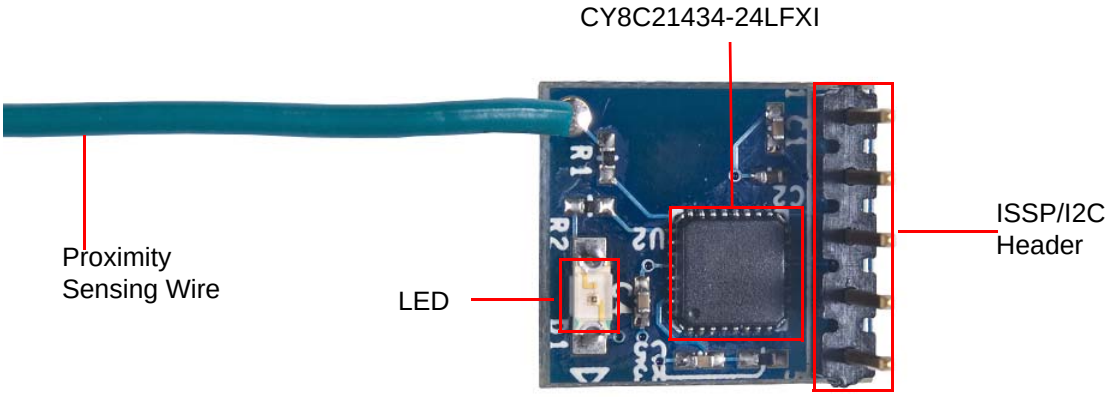
Figure 4-1. System Block Diagram




## 4.2 Functional Description

CY3235-ProxDet board includes PSoC CY8C21434, proximity sensor, indicator LED, and ISSP/I2C connector. Figure 4-2 shows the functional hardware components on the CY3235-ProxDet board.

Figure 4-2. CY3235-ProxDet Functional Components

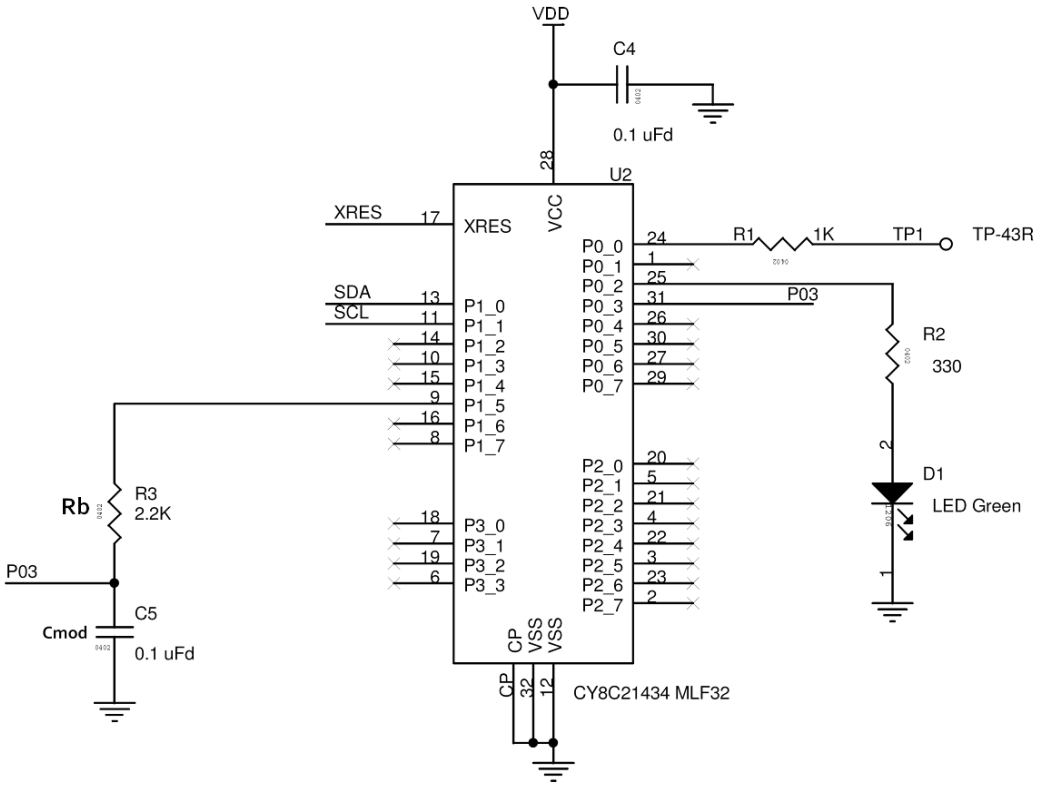



### 4.2.1 PSoC CY8C21434-24LFXI

The PSoC CY8C21434 is initially factory programmed as a CapSense Controller with the control circuitry to work with the CY3235-ProxDet kit. CY8C21434 along with CSD technology demonstrates proximity sensing. The mapping of PSoC pins is listed in 4.2.1.1 Pin Description of CY8C21434-24LFXI.

PSoC CY8C21434 is programmed through ISSP using a MiniProg and the data acquisition and checking of the output is done using the I2USB Bridge.

Figure 4-3. Schematic View of PSoC CY8C21434



#### 4.2.1.1 Pin Description of CY8C21434-24LFXI

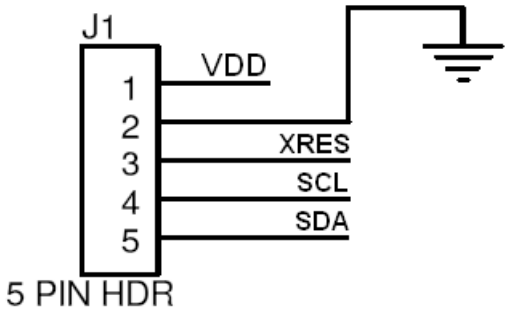
| Pin No. | Name | Description | Connected to |
|---|---|---|---|
| 1 | P0[1] | GPIO | NC |
| 2 | P2[7] | GPIO | NC |
| 3 | P2[5] | GPIO | NC |
| 4 | P2[3] | GPIO | NC |
| 5 | P2[1] | GPIO | NC |
| 6 | P3[3] | GPIO | NC |
| 7 | P3[1] | GPIO | NC |
| 8 | P1[7] | GPIO | NC |
| 9 | P1[5] | Bleed resistor (Rb) pin | R3 (Rb) |
| 10 | P1[3] | GPIO | NC |
| 11 | P1[1] | ISSP CLK, I2C SCL | J1 |
| 12 | Vss | Ground connection | |
| 13 | P1[0] | ISSP DATA, I2C SDA | J1 |
| 14 | P1[2] | GPIO | NC |
| 15 | P1[4] | GPIO | NC |
| 16 | P1[6] | GPIO | NC |
| 17 | XRES | Active high external reset with internal pull down | J1 |
| 18 | P3[0] | GPIO | NC |
| 19 | P3[2] | GPIO | NC |
| 20 | P2[0] | GPIO | NC |
| 21 | P2[2] | GPIO | NC |
| 22 | P2[4] | GPIO | NC |
| 23 | P2[6] | GPIO | NC |
| 24 | P0[0] | Proximity sensor pin | TP1 |
| 25 | P0[2] | LED (green) | D1 |
| 26 | P0[4] | GPIO | NC |
| 27 | P0[6] | GPIO | NC |
| 28 | Vdd | Supply voltage | |
| 29 | P0[7] | GPIO | NC |
| 30 | P0[5] | GPIO | NC |
| 31 | P0[3] | Integrating capacitor (Cmod) pin | C5 (Cmod) |
| 32 | Vss | Ground connection | |

### 4.2.2 ISSP/I2C Connector

In-system serial programmer (ISSP) is used to program the device using the MiniProg programmer and the USB cable. MiniProg can be plugged into the ISSP header J1.

The ISSP header J1 is also used to connect the I2USB Bridge for communication between the PC and the controller board. The pin mapping for the ISSP connector is shown in the following figure.

Figure 4-4. Schematic View of ISSP Connector



# 5. Code Examples

The code example is available at http://www.cypress.com/go/cy3235-proxdet.

## 5.1 Example_CY3235_CSD_PROX_Detector Project

### 5.1.1 Project Description

This project demonstrates CapSense proximity sensing by using a conducting wire as the sensor, PSoC device for detection, and LED for status indication. I2C communication is implemented in the PSoC device to monitor CapSense data of the proximity sensor. This includes raw counts, baseline, difference counts, and on/off status. The following user modules are used in this example.

| User Module | Hardware Resource allocation |
|---|---|
| CSD | 3 digital blocks and 3 analog blocks |
| EzI2Cs | Dedicated I2C block. Does not use any digital or analog block |
| LED | Software implementation. Does not use any digital or analog block |

The functionality of each user module is briefly explained here.

**CSD:** This user module measures capacitance and converts it into digital count (RawCount). It uses a robust sigma-delta algorithm for the capacitance to digital conversion. The CSD user module also provides high-level APIs that can be used to make high-level decisions in application firmware such as whether the sensor is activated or not.

**EzI2Cs:** This user module implements a register based I2C slave component in PSoC device. I2C slave supports up to 400 kHz speed of bit transfer rate. I2USB Bridge can be used as I2C master that provides an easy interface between the PC and the PSoC device.

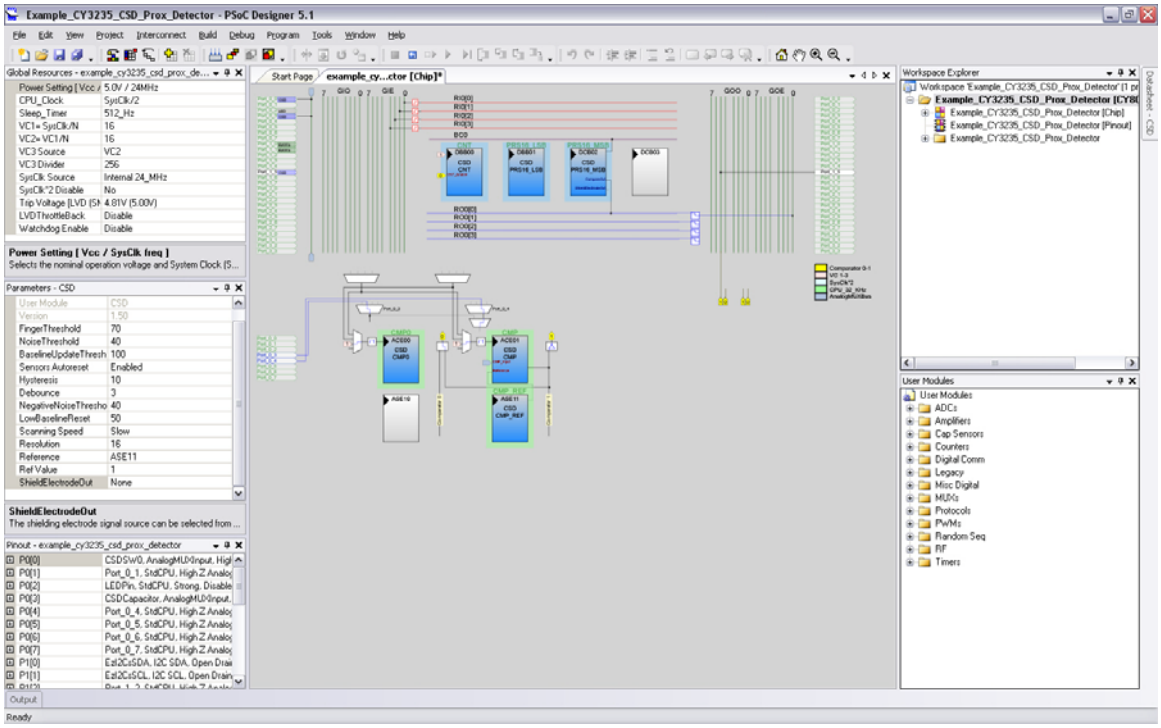
**LED:** This user module is implemented in software and does not use any hardware resource in PSoC. It enables a simple method of assigning PSoC pins for LED drive and provides APIs to control the LED.

PSoC Designer 5.1 SP1 version is used to create and test this code example. This example scans the proximity sensor continuously and drives the LED depending on its status. The EzI2Cs user module provides a register based access of CapSense parameters to the I2C master. In the firmware, a set of memory locations are defined as I2C buffer, which can be read by I2C master. This buffer contains the CapSense data and is updated after every sensor scan. I2USB bridge can be used as I2C master to read the CapSense data. Immediately after powering the device, instructions in *boot.asm* file are executed to initialize the hardware and invoke the main function. The main function initializes the CSD, EzI2Cs, and LED user modules. Then, the main function enters into a loop, which does the following:

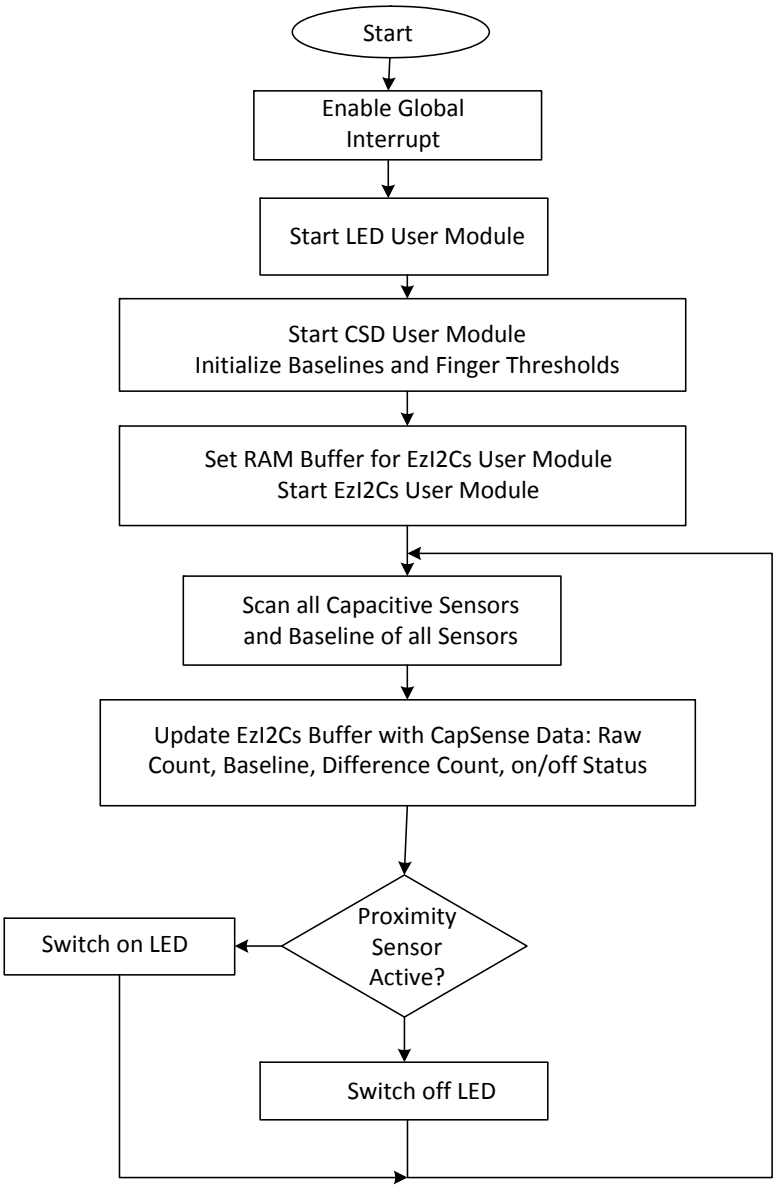
- Scans the proximity sensor
- Executes the baseline algorithm
- Stores the CapSense data in the I2C buffer
- Drives the LED based on the status of proximity sensor

## 5.1.2 Device Configuration

Figure 5-1. Device Configuration for CY3235_21434 CSD Proximity Project



### 5.1.3 Firmware Architecture

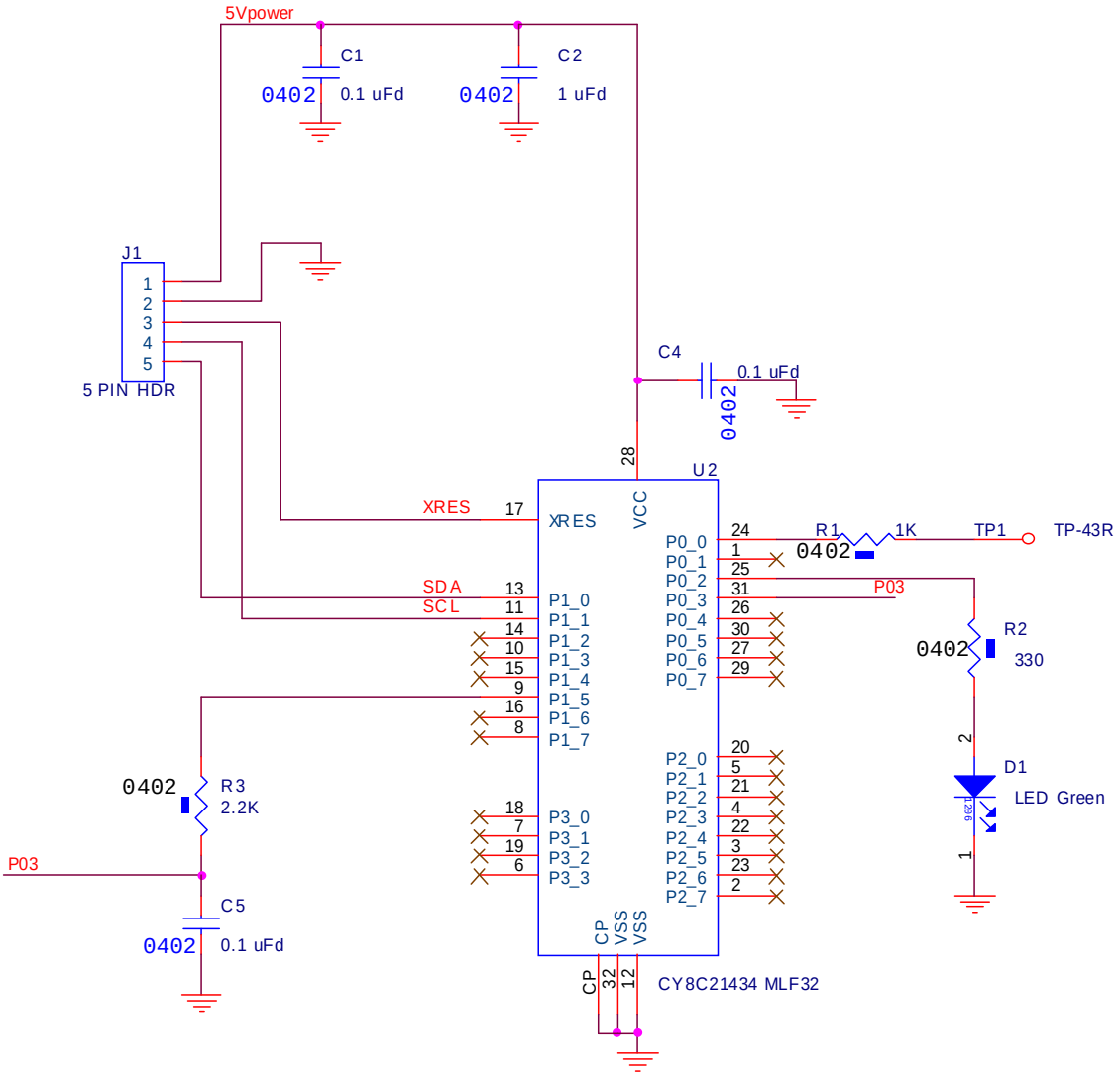



### 5.1.4 Verify Output

1. After downloading the project hex file into the device using MiniProg, connect I2USB Bridge to connector J1 and power the device.
2. Move your hand near the sensing wire and observe that the LED glows.
3. Follow the instructions in Bridge Control Panel on page 18 to get the chart view of the CapSense data in Bridge Control Panel.
4. Observe that the RawCount and Difference Count increase as you move your hand near the sensor. When the Difference Count exceeds Finger Threshold + Hysteresis limit, the status variable changes from zero to one and the LED glows.

# A. Appendix

## A.1 Schematic



## A.2 Board Layout

Figure A-1. Top View

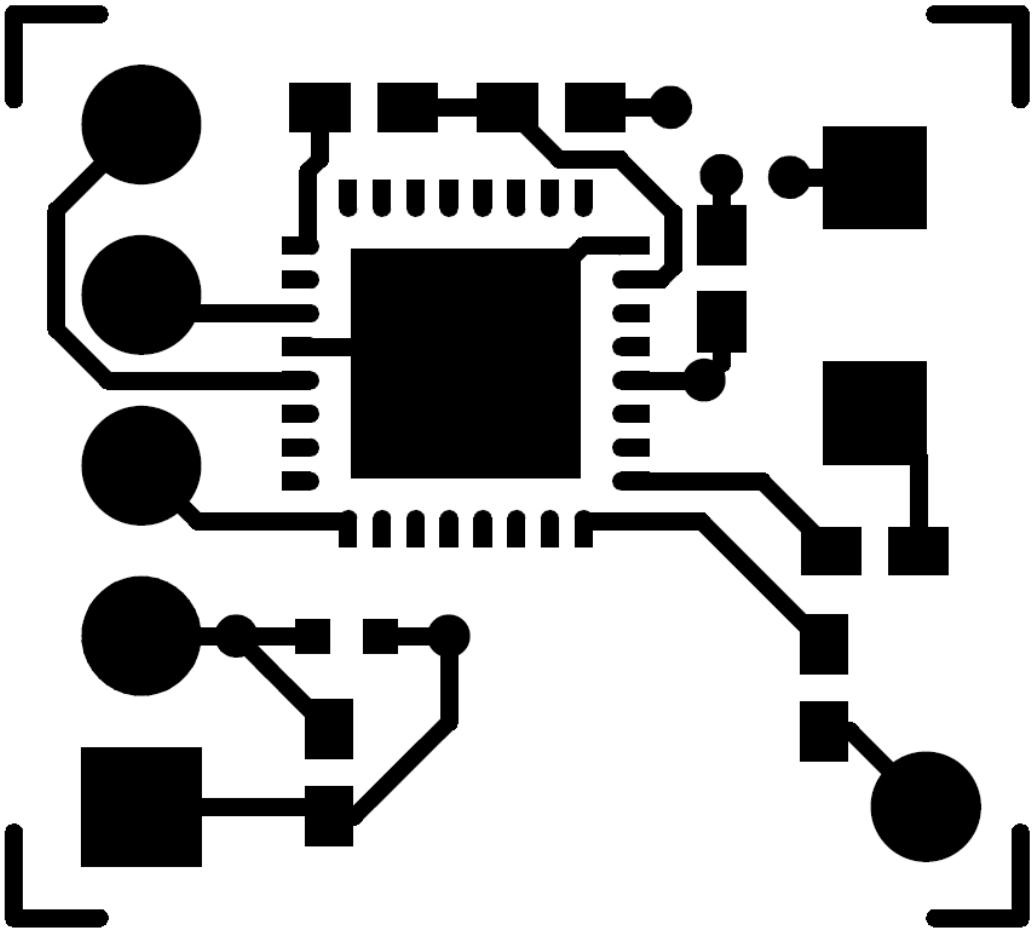
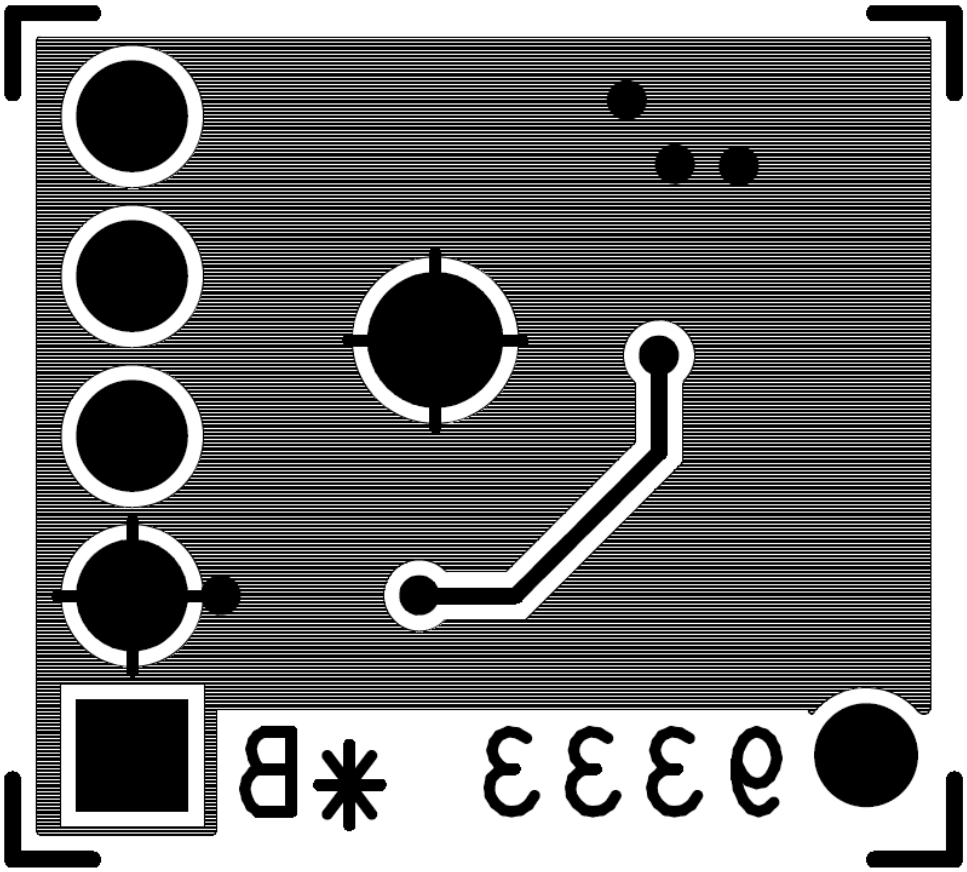


Figure A-2. Bottom View



## A.3 Bill of Materials (BOM)

| Item | Qty | CY Part Number | Reference | Description | Manufacturer | Mfr Part Number |
|---|---|---|---|---|---|---|
| | | PDC-9333 *B | | PCB | Cypress Semiconductor | PDC-9333 |
| 1 | 3 | 730R-10010 | C1,C4,C5 | CAP .1UF 16V CERAMIC Y5V 0402 | Panasonic - ECG | ECJ-0EF1C104Z |
| 2 | 1 | 730R-13400 | C2 | CAP 1 uF 6.3V CERAMIC X5R 0402 | Panasonic | ECJ-0EB0J105M |
| 3 | 1 | 860R-10292 | D1 | LED GREEN CLEAR 1206 SMD | Chicago Miniature Lamp, Inc | CMD15-21VGC/TR8 |
| 4 | 1 | 420R-13565 | J1 | CONN HEADER VERT 5POS .100 TIN | Molex/Waldom Electronics | 22-28-4050 |
| 5 | 1 | 610R-10752 | R1 | RES CHIP 1.0K OHM 1/16W 5% 0402 SMD | Phycomp USA Inc | 9C1A04021001JLHF3 |
| 6 | 1 | 610R-13651 | R2 | RES 330 OHM 1/16W 5% 0402 SMD | Yageo Corporation | RC0402JR-07330RL |
| 7 | 1 | 610R-11324 | R3 | RES 2.2K OHM 1/16W 5% 0402 SMD | Panasonic - ECG | ERJ-2GEJ222X |
| 8 | 1 | NA | TP1 | TEST POINT 43 HOLE 65 PLATED | NONE | |
| 9 | 1 | CY8C21434-24LFXI | U2 | IC PROGRAMMABLE SOC MLF32 | Cypress Semiconductor | CY8C21434-24LFXI |
| 10 | 1 | | | 6" 24 gauge stranded wire | | |
| **Special Installation Components** | | | | | | |
| 11 | | | | Strip end of wire and solder to TP1 | | |